

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A) 平2-15381

⑮ Int. Cl.³

G 06 F 15/66

識別記号

4 5 0

庁内整理番号

8419-5B

⑬ 公開 平成2年(1990)1月19日

審査請求 未請求 請求項の数 27 (全32頁)

⑭ 発明の名称 ラスタ操作実行方法、時間領域多重化方法および画像処理方法

⑰ 特 願 平1-71604

⑱ 出 願 平1(1989)3月23日

優先権主張

⑳ 1988年3月23日㉑ イギリス(G B)㉒ 8806837

㉓ 1988年3月23日㉔ イギリス(G B)㉕ 8806845

⑲ 発 明 者

ネイル フランシス
トレヴェット英国 ケイティー2 7エイエル キングストン アボン
テムズ マヌーアゲート ロード 16

⑲ 発 明 者

マルコム エリック
ウイルソン英国 ドーセット州 ブリッドポート サルウエイアツシ
ユ サルウエイアツシユ トライヴ 16

㉖ 出 願 人

デュボン ピクセル
システムズ リミテッ
ド英国 エスジーキュー 4キューエヌ ハートフォードシ
ヤ州 スティーヴネイジ ウェツジウッド ウエイ(番地
なし)

㉗ 代 理 人

弁理士 谷 義 一

明 細 書

1. 発明の名称

ラスタ操作実行方法、時間領域多重化方法
および画像処理方法

2. 特許請求の範囲

1) バッチ形式化した画素データについてラスタ
操作を実行する方法であって、

第1バッチを得るステップと、

第2バッチを得るステップと、

前記第1バッチの予め選択された画素と前記第

2バッチの予め選択された画素とをマーキングし
て、前記第1および/または第2バッチと同じ数
の画素を有するシフトしたバッチを生成するステ
ップと

を具えたことを特徴とするラスタ操作実行方法。

2) 前記第1および第2バッチを、前記バッチの
予め選択した列から予め選択した画素と同じバツチのラインから得ることにより、前記シフトした
バッチを、前記ラインの方向(X方向)にシフト
されているものとしたことを特徴とする請求項1
記載のラスタ操作実行方法。3) 前記第1および第2バッチを、前記バッチの
予め選択した行から予め選択された画素と異なる
バッチのラインから得ることにより、前記シフト
したバッチを、前記ラインを横切る方向(Y方
向)にシフトされているものとしたことを特徴と
する請求項1記載のラスタ操作実行方法。4) 第1バッチA1および第2バッチA2に関し請求
項2記載の方法を実行して、X方向にシフトした
バッチを生成し、前記第1バッチA1および前記第2バッチA2とは
異なるラインにある第1バッチB1および第2バツ
チB2に関し請求項2記載の方法を実行して、X方
向にシフトしたバッチを生成し、

前記X方向にシフトしたバッチAXおよび前記X

方向にシフトしたパッチBXに関して請求項3に記載の方法を実行して、XY方向にシフトしたパッチを得ることを特徴とするラスタ操作実行方法。

5) 前記シフトしたパッチと前記第1および第2のパッチのいずれかとの間で論理および/または算術演算を実行するステップをさらに具えたことを特徴とする請求項1ないし4のいずれかの項に記載のラスタ操作実行方法。

6) 第3パッチを得るステップと、

前記シフトしたパッチと前記第3パッチとの間で論理および/または算術論理演算を実行するステップと

をさらに具えたことを特徴とする請求項1ないし4のいずれかの項に記載のラスタ操作実行方法。

7) 前記第1および第2パッチはソースパッチで

12) ライン上のすべてのパッチが処理されるまで請求項1ないし11のいずれかの方法を繰返し実行することを特徴とするラスタ操作実行方法。

13) 2以上の平面を有するパッチ形式化された画素データについてラスタ操作を実行するラスタ操作実行方法であって、

第1パッチの第1平面を得るステップと、

第2パッチの第1平面を得るステップと、

第1パッチの第1平面内の予め選択したビット群と、第2パッチの第1平面内の予め選択した他のビット群とをマージし、第1および/または第2パッチの第1平面として同じ数のビット群を有するシフトされたパッチ平面を生成するステップと

を具えたことを特徴とするラスタ操作実行方法。

14) 前記第1パッチの第1平面は、前記第1パッチを得てそこから第1平面を得ることによって得

あり、前記第3パッチはデスティネーションパッチであることを特徴とする請求項6記載のラスタ操作実行方法。

8) 前記XY方向にシフトしたパッチの部分をデスティネーションに書込むステップをさらに具え、前記部分は書込みマスクに関して選択されていることを特徴とする請求項4記載のラスタ操作実行方法。

9) 前記書込みマスクは画像平面または画像平面のパッチ平面に設けられることを特徴とする請求項8記載のラスタ操作実行方法。

10) 前記画像平面はソース平面であることを特徴とする請求項9記載のラスタ操作実行方法。

11) 前記画像平面はデスティネーション平面であることを特徴とする請求項9記載のラスタ操作実行方法。

られ、および/または前記第2パッチの第1平面は、前記第2パッチを得てそこから第1平面を得ることにより得られることを特徴とする請求項13記載のラスタ操作実行方法。

15) 前記第1および第2パッチのそれぞれの平面について請求項13または14記載の方法を繰返し実行することを特徴とするラスタ操作実行方法。

16) 前記シフトされたパッチ平面は同じ数のビット群を有することを特徴とする請求項15記載のラスタ操作実行方法。

17) 2以上の平面を有するパッチ形式化された画素データにラスタ操作を実行するための時間領域多重化方法であって、

第1クロックサイクルで第1パッチを得るステップと、

前記第1クロックサイクルの第1時間部分で前記第1パッチの第1平面を選択するステップ

と、

第2クロックサイクルで第2バッチを得るステップと、

前記第2クロックサイクルの第1時間部分で前記第2バッチの第1平面を選択するステップと、

前記第1バッチの前記第1平面内で予め選択した部分と、前記第2バッチの前記第1平面内で予め選択した部分とをマーキングし、シフトされたバッチの第1平面を生成するステップと

を具えたことを特徴とする時間領域多重化方法。

18) 前記第1クロックサイクルの第2時間部分で前記第1バッチの第2平面を選択するステップと、

前記第2クロックの第2時間部分で前記第2バッチの第2平面を選択するステップと、

前記第1バッチの前記第2平面内で予め選択した部分と、前記第2バッチの前記第2平面内で予

法であって、

バッチの第1平面内でいくつかの数の画素成分を選択するステップと、

バッチの第2平面で前記数の画素成分を選択するステップと、

前記第2平面内で選択した画素成分と前記第1平面内で選択した画素成分とを置換えるステップと
を具えたことを特徴とする画像データ処理方法。

22) 前記各平面において前記画素成分のすべてが選択されることを特徴とする請求項21記載の画像データ処理方法。

23) 画素成分の2次元アレーが選択されることを特徴とする請求項21記載の画像データ処理方法。

24) 前記第1および第2平面は同一バッチ上の平面であることを特徴とする請求項21ないし23のい

め選択した部分とをマーキングし、シフトされたバッチの第2平面を得るステップと
をさらに具えたことを特徴とする請求項17記載の時間領域多重化方法。

19) 前記予め選択した部分は予め選択した列であり、前記シフトされたバッチはX方向にシフトされたバッチであることを特徴とする請求項17または18記載の時間領域多重化方法。

20) 前記予め選択した部分は予め選択した行であり、前記シフトされたバッチはY方向にシフトされたバッチであることを特徴とする請求項17ないし19のいずれかの項に記載の時間領域多重化方法。

21) 画像が複数のバッチによって表現され、それぞれのバッチが2次元の画素のアレーを表現し、各画素のデータは各平面におけるそれぞれの画素成分を有するものである前記画像データの処理方

ずれかの項に記載の画像データ処理方法。

25) 前記第1および第2平面は異なるバッチ上の平面であることを特徴とする請求項21ないし23のいずれかの項に記載の画像データ処理方法。

26) 前記異なるバッチは異なる画像に関連していることを特徴とする請求項25記載の画像データ処理方法。

27) 前記第1平面内で選択した画素成分と前記第2平面内で選択した画素成分を置換えることにより、前記第1平面と前記第2平面との間で前記選択した画素成分を交換するステップをさらに具えたことを特徴とする請求項21ないし26のいずれかの項に記載の画像データ処理方法。

(以下余白)

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は、概して画像処理に関し、特に画像メモリ内でラスタ操作を実行するための装置および方法に関するものである。

〔従来の技術〕

ラスタ操作は、画像の四角形部分の操作を行うために使用される。画像メモリ内の第1四角形領域（ソース）および第2四角形領域（デスティネーション）は、相互間でいかなるブール演算でも実行することができる。ブール演算の結果は、デスティネーション四角形領域に位置付けられる。最も単純なラスタ操作は、単にソース画素をデスティネーション画素に置換することである。これは通常、画素対画素を基にして実行される。各々の画素にはアドレスが割り当てられており、読み込み画素は、新しいアドレスに従って、デスティネーション領域の新しいロケーションへ移行する。このような移行は、通常コピーと呼ばれ

ンでの二つの画像の合成は、画像マージのひとつである。1ビットの画像では、2つの画像は、 'OR' 演算を使用してマージされる。両画像のバックグラウンドが '0' で、値 '1' の画素によって画像内のオブジェクトが表されるとすれば、論理 'OR' は、デスティネーション画像に両画像のすべてのオブジェクトが含まれることになる。

論理演算 'OR' を使用したブロックコピー操作の一例を第1図および第2図により説明する。ブロックコピーされる第1画像を含むソース102と、第2画像を含むデスティネーション領域104で、論理演算 'OR' を実行すると仮定する。ソースから、デスティネーションへコピーを実行している間に、2領域102、104の論理和が実行され、その結果、デスティネーション領域104に双方の画像が現される（第2図）。ソースとデスティネーションの領域間で論理演算が実行されるようなブロックコピー操作では、デスティネーション領域が論理演算 'OR' のデータを提供するため

ている。

単純なコピーにはブロック転送処理が含まれる。ソース・ロケーションの画素は、グループとして所望のデスティネーション・ロケーションへコピーされる。典型的なブロック転送は、画素対画素を基に実行される。このようなコピーの結果、ディスプレイ画面上でのソース・ロケーションからデスティネーション・ロケーションへと画像が移動する。

さらに高度なラスタ操作には、通常、移動した画素のブール演算が含まれており、デスティネーション・ロケーションの画素はブール演算の結果を反映する。ブール演算により、コピーされた画像に所望の変更を加えることができる。周知の通り、ブール演算は画素上で実行される論理演算である。ブール演算は桁の切り上げや切り下げを必要としないため、これらを必要とする算術演算（加減乗除）と、対照的である。

従来のブール演算の一般例としては、画素のマージがある。デスティネーション・ロケーシ

の第二ソースとして有効的に働くということが理解されるべきである。

以下の文献には、一般的なラスタ操作についての優れた考察が述べられている。本願には、その全文が参考として統合されている。

- ・ Section 5-6 "Raster Methods For Transformations" : Hearn, Donald and M. Pauline Baker : Computer Graphics. (Prentice-Hall International, 1986)
- ・ Chapter 5 "Segments" : Harrington, Steven; Computer Graphics-A programming Approach. (McGraw-Hill Inc., 1983, International student Edition)
- ・ Chapter 5 "Clipping and Windowing", Chapter 15 "Raster-Graphics Fundamentals", Chapter 19 "Raster Graphics Systems", Chapter 18 "Raster Display Hardware" ; Newman, William M. and Robert F. Sproull : Principles of Interactive Computer Graphics

(McGraw-Hill International Book Company,
1981, International Student Edition)

[発明の概要]

本発明は、画像メモリ（ディスプレイ・メモリおよびもしあればノンディスプレイまたはオフスクリーン・メモリを含む）内で、ラスタ操作ブロック転送機能を実行するための新しくかつ効率的なシステムおよび方法を具えるものである。種々の実施例において、操作には単純コピー、すべてのブール関数（OR, AND, INVERT, NO OPERATION, XOR およびこれらの組み合わせを含むが、この限りではない）、すべての算術関数（ADD, SUBTRACT, MULTIPLY, DIVIDE およびこれらの組み合わせを含むが、この限りではない）、平面スワッピングおよび／またはマスク・コピーを適用することができる。

画素対画素のアプローチとは反対に、ソースとデスティネーション（領域またはアドレス）間のバッチ・アプローチが使用される。バッチ・アプ

チ（および指定可能ロケーション）を持っていることを意味している。画像メモリと関連ディスプレイモニタのアドレス指定が同次元である事は、概念上多くの操作を簡易化し、実行速度を高める。

バッチ・アクセス処理の欠点は、このシステムがスピードと概念上の簡易性を得た一方で粒状度（granularity）を損失したことにある。すなわち、1バッチより小さいデータグループの操作、およびバッチ境界内に定義されていない画像領域での作業が難しくなったということである。

本発明のシステムおよび方法は、1画素までの粒状度の画像を操作し、グループの画素データをバッチ境界内に定義されていない領域間でコピーするという点で、バッチ・アクセス・プロセッサの能力を著しく改善した。

本発明者らは、多くの重要な画像処理を1画素の粒状度およびアドレス指定能力で実行可能とし、バッチ・アクセス処理の全ての利点を得るようなシステム・アーキテクチャと操作方法を発明

ローチは、画素が、一つのメモリサイクルの（予め選択された隣接画像の四角形領域の）画素の2次元バッチの画像（フレーム）メモリからアクセスされるという事実と関連している。

バッチ・アクセス・システムおよび方法には、画像データを大きなグループ（例えば、20画素のバッチ）で処理できるという利点がある。従って、画像の処理、表示および保存のスピードを著しく速めることができる。例えば、8ビット画素5×4の配列により定義されるバッチをアクセスする場合、フレーム・メモリは、同時に160ビットの読み込みおよび書き込みを実行することができる。すなわち、画素データバスのバンド幅を画素アクセスシステム（すなわち、画素は8ビットにより定義される）の20倍に増加することができる。さらに、5×4バッチ構造は、1280×1024画素の標準的な高解像度グラフィック・ディスプレイ・モニタに良好に適合する。5×4バッチの画素データ構造はスクリーン・リフレッシュ・メモリが、X、Yの両方向にそれぞれ正確に256バツ

した。さらに、本発明者らは、ビット・ポジションを同じ画素内で交換および／または異なる画素間でスワップさせることによって、画素データをビット平面内のいたる所で処理するシステムおよび方法を確立することにより、従来の画素粒状度の限界という概念を超越した。

[実施例]

以下、図面を参照し、次の目次に従った順序で本発明を詳細に説明する。

(目次)

- I 概 要
- II 本発明のシステムおよび方法の好適実施例
 - a) システム環境
 - b) 画像データ形式
 - c) 操作の概略
 - d) 入力バッチ・レジスタ
 - e) シフトおよびマージ操作の目的
 - f) Xシフトおよびマージ
 - g) 時間領域の多重化 (Time Domain Multiple-

xing)

h) Yシフト

i) ライン・ストレージRAM

j) 論理装置

k) 出力マルチプレクサ

l) 出力バッチ・レジスタ

m) 書き込み(ライト)マスク

n) ページ・モード・アドレッシング

o) シフト計算

III 平面スワッピングおよびビット・ポジション
処理

IV 変形と強調

V 結 論

I 概 要

広義的に、本システムおよび方法は、画像記憶装置とは別に、シフト論理および非表示RAM領域を採用している。シフト論理は、コピーする画像のバッチのシフトおよび水平マージに使用される。非表示RAM領域(ライン・ストレージRAM)は、垂直マージおよび画像記憶装置のデスティネーション・ロケーションにコピーする画像の全ラインを一時的に保存するために使用される。

保存された画像を画像記憶装置の画像のデスティネーション・ロケーションに合成するため、論理ユニットもまた使用されている。特定のプール関数または算術関数を使用することにより、画像記憶装置に記憶されている一つ以上の画像領域を、同時に処理することができる。

II 本発明のシステムおよび方法の好適実施例

第3図は、本発明のシステムおよび方法の好適実施例を示す概要ブロック図である。

a) システム環境

第3図に示すように、画像メモリ302は、ビデオ・ディスプレイ装置(不図示)の画素ロケーションに相当する2次元メモリ・ロケーション・フォーマットに従って、画素単位でビデオ画像を記憶している。画像メモリ302は、ダイナミックRAMのようなランダム・アクセス記憶装置を使用することにより、いかなる設計も可能である。好適には、画像記憶装置は、日本の日立社のmodel 53462 VDRAMのようなビデオRAM(VRAM)を使用した従来の設計のものである。本実施例のシステムで使用している画像メモリは、非選択平面をマスキングしながら、選択した画像平面の部分集合を書き込む能力を持っている(すなわち、各画素を定義するビットの部分集合を書き込む)。さらに、メモリは、X方向のVRAMのページ・モード・アクセスを実行できるようになっているのが好ましい。画像メモリとしては、英国5 Penrhyn Road, Kingston upon Thames, Surrey KT1 2BTにあるDu Pont Pixel Systems Limited(旧名 bench-

Mark Technologies Limited)製のDu Pont Pixel Systems bfs Framestoreを好ましく用いることができる。これら実施例で使用する装置は代表に過ぎず、既存あるいは今後開発されるすべての適合性のあるRAM装置が、本発明に適用できるということが理解されなければならない。

ビデオ・ディスプレイ装置(図には示されていない)は、画像メモリ302に記憶されたビデオ画像を表示する。本実施例のビデオ・ディスプレイ装置は、日本の東京にあるソニー社製のmodel GTM 1901-22のような高解像度ラスタ走査ビデオ・モニタである。このモニタは、横方向(行)1280画素、縦方向(カラム)1024画素の2次元ビデオ画像を表示するものであり、本発明の実施例に適合する。既存あるいは今後開発されるすべての適合性のあるディスプレイ装置も本発明に適用できる。

画像メモリ302は、出力ポート304を備えており、データ・バス306と接続している。画像メモリ302は、アドレス/コントロールバス310によ

り供給されるプロセッサ308からのアドレス信号に
 応答して、隣接メモリ・ロケーションに記憶さ
 れている画素をデータ・バス306に出力する。プ
 ロセッサ308は以下に説明する本発明の様々な段
 階(ブロック)の処理を制御する。本実施例で
 は、プロセッサ308は、ビット・スライス・グラ
 フィックス・プロセッサで、本発明の様々な段階
 を命令マイクロコードを使用して制御する。
 プロセッサとしては、Du Pont Pixel Systems
 Limited、(英国,5 Penrhyn Road,Kingston upon
 Thames, Surrey KT1 2BT)のDuPont Pixel
 Systems GIPを好ましく用いることができるが、
 本発明がその他のプロセッサも適用可能とするこ
 とを理解されたい。

b) 画像データ形式

本発明の一形態では、画素データを出力、操
 作、処理および供給するために、バッチ処理を使
 用している。バッチは、画像メモリ302に記憶さ
 れている画像の隣接する画素の1グループであ

る。

本例のバッチ構造は、第4図を参照すると理解
 しやすい。第4図は、404,406,408,410,412,414,
 146,418の8平面から成る5×4バッチ(包括し
 て参照番号402で表される)を示している。各平
 面は、バッチ402に定義されている20画素の1ピ
 ットの深さのスライスである。本例のバッチ402
 は、各画素の定義に8ビットを使用するため、各
 バッチは8平面から成っている。仮に、各画素を
 1ビットで定義するとすれば、バッチは1平面し
 た持たないということになる。通常、1バッチ
 は、その画素を定義しているビット数と同数の平
 面から成っている。このため、バッチの各平面
 を、平面またはバッチ平面と呼ぶことにする。

第4図はまたバッチおよびバッチ平面の画素デ
 ータの番号構成を示している。第4図から、定義
 されたバッチは4行5列であることが分かるであ
 ろう。バッチおよびバッチ平面内の画素ポジシ
 ョンには、0(左下)から19(右上)までの番号が
 付けられている。これらの番号は、本書の参照番

る。バッチの最少単位は、2行2列の4画素から
 成る正方形である。最大バッチは、ディスプレイ
 装置のディスプレイ・フレームを形成している画
 素の総数となる。

本実施例で使用しているバッチは、第4図に示
 すような横5画素、縦4画素の長方形である。従
 って、本例のバッチは、隣接する20画素で形成さ
 れている。各画素にはビット番号が指定されてお
 り、データワードを構成している。画素に指定さ
 れたデータワードには、デジタル語で表現可能
 なデジタル状態番号で定義された連続体(パレ
 ット)の画素の色、彩度などの指定情報が含まれ
 ている。本実施例では、画像メモリ302に記憶さ
 れた各画素ワードは、8ビットである。しかしワ
 ードのサイズは、内容の範囲および/または画素
 の記憶に必要な一定の機能/条件により、1以上
 のいかなるビット数にもなり得ることが理解され
 なければならない。従って、本例の5×4画素の
 バッチには160ビットのデータが含まれているこ
 とが分かるはずである(5画素×4画素×8ピッ

トとして使用される。例えば、バッチ平面のピッ
 ト7とは、(概念上)下から2行目、左から3列
 目の位置にあるビットを示している。ビットの番
 号構成は、本書の20ビット・バッチ平面データ
 ・バスにも同様に供給されるものとする。

また、本書では、バッチ平面の行と列とが定義
 されている。バッチ平面の第1行(行1)は、
 0, 1, 2, 3, 4のデータ・ポジションから成
 ると定義される。第2行(行2)は、5, 6,
 7, 8, 9のデータ・ポジションから成ると定義
 される。第3行(行3)は、データ・ポジション
 10,11,12,13,14から、第4行(行4)は、15,16,
 17,18,19から成ると定義される。バッチ平面の第
 1列(列1)は、0, 5, 10,15のデータ・ポジ
 ションから成っている。第2列(列2)は、1,
 6, 11,16のデータ・ポジションから成ってい
 る。第3列(列3)は、2, 7, 12,17のデータ
 ・ポジションから成っている。第4列(列4)
 は、3, 8, 13, 18のデータ・ポジションから
 成っている。第5列(列5)は、4, 9, 14,

19のデータ・ポジションから成っている。

上記内容から、ビデオ・ディスプレイ画面および画像メモリは、 5×4 バッチ（またはその他の与えられたバッチ次元）の複数の水平行から形成されたものとして概念化することができる。同様に、すべての四角形画像領域は、これらのバッチの水平行から形成されていると概念化できる。

本書では、バッチの行は“ライン”または“バッチ行”として呼ぶことにする。この定義は、“ライン”（上記で定義したバッチの行）と、以降単に“行”と呼ぶバッチ内の画素の行（またはバッチ平面のビット群）との相違を明確にするためのものである。本書で使用される“カラム”という語は、バッチ内の画素の列（またはバッチ平面のビット群）を意味する。

また、本書では、画像を形成しているすべてのバッチの1ビットの深さのスライスを、画像平面と呼ぶことにする。

は2つの目的に使用される。第1目的は、現行および既存の入力バッチ平面の行から選択された画素の行をマージして、完全な出力バッチ平面を作成することである。第2目的は、ページ・モードの画像メモリ302がデータを直接読み込むことができるよう、完全にシフトおよびマージされたバッチ平面の全ラインを記憶することである。

ライン・ストレージRAMから、新しく作成されたバッチ平面が論理ユニット317に転送される。論理ユニットは、新しく作成されたバッチ平面とデスティネーション・バッチ・データ（平面の一つが論理ユニット入力端子に適切に準備されているもの）との間で、ブール演算を実行する。論理ユニット317からは、バッチ平面にいくつかの処理が行われる。第1に、バッチ平面は、グラフィックス・プロセッサ308の書き込みマスク・レジスタ320を起動させるのに用いることができる。第2に、各平面は、2:1 MUX 324の操作を通じて第2データ・バス322のもう一つのラス

c) 操作の概略

本装置および方法の実施例では、画素データのバッチは、グラフィックス・プロセッサ308により供給されるアドレスおよびコントロールデータに応じて画像メモリ302からアクセスされる。入力バッチ・レジスタ312へは、1回に1バッチずつロードされる。入力バッチ・レジスタは選択された各バッチの平面を、Xシフトおよびマージ論理回路314および論理ユニット317の1入力端子へ出力する。Xシフトおよびマージ回路内では、バッチ・データの平面は、X方向に所望の数値だけシフトされ、次に、水平に隣接したバッチのデータ平面とマージされる。

Xシフトされたバッチ平面は、Xシフトおよびマージ論理回路からYシフトブロック316へ転送される。Yシフト論理回路は、バッチ平面をY方向に所望の行数だけシフトするが、マージは実行しない。XおよびYシフトされたバッチ平面は、次に中間RAM（すなわちライン・ストレージRAM 318）に保存される。ライン・ストレージRAMに

操作プロセッサからの平面と置換することができる。2つの3状態バッファ326, 328は、他のラス

タ操作プロセッサと現システムとの間のマスク・データの流れを制御する。最後に、平面は2:1 MUXを通じて出力レジスタ330に転送され、すぐに画像メモリに書き込まれる。この処理はコピーするバッチ平面が残っている限り繰り返される。

できれば、完全なソース四角形が、一度に一つの（非TDM）または二つの（TDM）画像平面の処理によって処理され、次の画像平面の処理に戻るのが望ましい。表示破壊作用を低減するためには、最も重要な画像平面を最初に処理するのが好ましい。本装置および方法において、最も重要という用語は、表示画像の完全性に最も影響を与える画像平面を意味している。通常、最も重要な画像平面は、各画像バッチ内の高配列ビットの画素を占めている。

明確化のため、本装置を以降Blit（ブロック転送）プロセッサと呼ぶことにする。

d) 入力バッチ・レジスタ

入力バッチ・レジスタの目的は、各々の完全なバッチに関連する160ビットのデータを集積し、Blitプロセッサに処理する一つの平面（すなわち20ビット、1画素1ビット）を選択させることである。本実施例では、完全なデスティネーション（転送先）画像データが作成されるまで、一度に一つ（非TDM）または二つ（TDM）の平面上で、ラスタ操作が実行される。この処理を達成するために、入力バッチ・レジスタ312は完全な160ビット・バッチを保存し、グラフィックス・プロセッサ308からの制御データに応じて選択された処理平面を出力する。

入力バッチ・レジスタ（第9図）は、8個の20ビット・レジスタ902, 904, 906, 908, 910, 912, 914, 916から成っており、各レジスタは、可逆データ・バス306上のバッチ・データの1平面を受信する。レジスタを起動したい場合は、グラフィックス・プロセッサ308が論理回路ゲート922に接続する入力レジスタロードイネーブル^{*}ライン91

入力0レジスタ・フェーズ1入力平面選択ライン926（PAL 924にTDM Blitプロセッサ操作の第1第図フェーズに与える平面を選択させるためのもの）；3本のフェーズ2入力平面選択ライン928（PAL 924にTDMラスタ操作の第2フェーズに与える平面を選択させるためのもの）；PALにシングル・フェーズ（非TDM）と、デュアル・フェーズ（TDM）とのどちらが実行されるのかを伝達するデュアル^{*}ライン930；PAL 924にフェーズ1とフェーズ2の平面選択入力を正確に選択させるためTDMのどのフェーズを実行させるのかを伝達するフェーズ1^{*}ライン932である。プロセッサ・クロック1000のタイミング、すなわちフェーズ1^{*}信号1006は第10図に示されている。

非TDM（1平面）操作では、デュアル^{*}ライン930は常にハイ、フェーズ1^{*}ラインは常にローとなっている。これによりPAL 924は、常にフェーズ1平面選択入力に指定された平面を選択する。TDM操作では、フェーズ1^{*}ラインが循環している間デュアル^{*}ライン930は常にローとなってい

（左下）に出力を行う。これにより、プロセッサ・クロック1000（ライン920上）が、可逆データ・バス306からの160ビット・バッチを持つ入力レジスタを起動する。プロセッサ・クロック1000は、グラフィックス・プロセッサ308により生成され、画像メモリ・データバス306に接続する画像メモリ302からバッチが出力されるごとに1回循環する。第9図に示すように、画像メモリ・データ・バス306の全160ビットのバッチは、入力レジスタに平行にロードされる。バッチ・データがレジスタにロードされると、Blitプロセッサの操作が開始可能となる。

PAL 924は、入力レジスタの出力使用可能ラインの制御を行い、Blitプロセッサにより処理される平面を制御している。8平面（20ビット）のどの平面を出力するかを選択するために、PALは、8本のコントロールラインを使用する。これらのラインは、グラフィックス・プロセッサ308により生成されるのが望ましい。これらは、以下のラインから構成されている。すなわち、3本の

る。これによりPAL 924はフェーズ1^{*}信号1006（ライン932）がローのときはフェーズ1平面選択入力により指定された平面を選択し、フェーズ1^{*}信号がハイのときは、フェーズ2平面選択入力により指定された平面を選択する。

入力レジスタ・ロードイネーブル^{*}信号ラインがローのとき、バッチ平面は対応するレジスタに記録される。選択されたバッチ平面のレジスタはPAL 924により出力可能となり、処理が実行される。

e) シフトとマージの目的

画素を、バッチ・アクセス画像メモリ302（本好適実施例の画像メモリ）のラスタ操作ブロック転送で配列するためには、2方向（XおよびY）へのシフトおよびマージが実行可能でなければならない。バッチ・アクセス・プロセッサでは、画素は2次元バッチの一郎として画像メモリからアクセスされる。ラスタ操作では画素の操作が必要なため、ソース・バッチは、ソース画素が要求さ

れたデスティネーション画素に対応できるよう2次元でシフト可能でなければならない。また、シフト操作により出力画素は正確なポジションに移動するが、シフトにより作成された出力パッチは、パッチ内の画素がすべて同じデスティネーション・パッチに属すとは限らないため、直接画像メモリに書き込むことができない。従って、画像メモリに保存できるパッチを作成するために、シフトしたパッチのマージが必要となる。

シフトおよびマージ操作の必要性は、第5図を参照すると理解しやすい。この図は、4つのパッチ(502, 504, 506, 508)から成る画像メモリ302のソース区域を示している。各パッチは、5×4配列の画素データを持っている。パッチ境界上にはない(太線で囲まれた)画素データ・グループ510を、デスティネーション領域512のパッチにコピーすると仮定する。典型的なパッチ・アクセス・プロセッサ内の画像メモリは、パッチ境界の粒状度までしか指定可能ではないため、プログラマは問題に遭遇する。

ンの完全な1行を処理する。既述したように、Blitプロセッサは一回につきパッチ・データの1平面のシフトおよびマージを実行する。従って、5×4パッチ平面の各行には、5ビットの情報が含まれている。第1第図のバレル・シフター608は、パッチ・レジスタ312から出力された0から4まで(行1)のビットを受信する。第2のバレル・シフター606は、5から9まで(行2)のビットを受信する。第3のバレル・シフター604は、10から14まで(行3)のビットを受信する。第4のバレル・シフター602は、15から19まで(行4)のビットを受信する。バレル・シフターは、5ビットXシフト制御バス810上のデータに基づいて、パッチ・データの行を順番に左または右へ該当数分シフトする。デコーダPAL 632は、このデータをバレル・シフターが使用する3ビットのシフト制御データにデコードする。各バレル・シフターのシフト制御入力端子は、デコーダから出力される3ライン634を受信する。デコーダPAL 632については後述する。

この問題の解決方法の一つは、4つのパッチ502, 504, 506, 508の全データを読み込み、セクション510のデータを持つ新しいパッチにマージすることである。マージされたデータは、可逆バス306を通じて画像メモリ302のデスティネーション領域512に書き込まれる。Blitプロセッサはページ・モードでシフトおよびマージを実行(すなわち一度にパッチの一行を処理)するため、全体の操作時間が短縮される。

本例のシフト論理回路は、すべて5×4パッチを処理するように設計されている。従って、特記事項がない限り、読者は5×4パッチまたはパッチ平面が処理されていると仮定されたい。

f) Xシフトおよびマージ

Xシフトおよびマージ論理回路314は、第6図を参照すると理解しやすい。第6図に示すように、Xシフトおよびマージ論理回路は、4つの5ビット・バレル・シフター602, 604, 606, 608を備えている。各バレル・シフターは、各パッチ・プレー

Xシフト・データの各行は、バレル・シフターから4つの5ビット・レジスタ612, 614, 616, 618の一つにクロックに同期して記録される。各バレル・シフター602, 604, 606, 608は、シフトデータを受信する5ビット・レジスタ612, 614, 616, 618の一つと対応している。

5ビット・レジスタへのデータの記録は、ライン620上のグローバル・Blitクロック1004により制御される。Blitクロック1004(第10図)は、概してBlitプロセッサの動作を制御している。プロセッサ・クロック1000が循環するごとに、新しいデータ・パッチが入力パッチ・レジスタ312に記録される。Blitクロック1004が循環するごとに、パッチの選択された平面が入力パッチ・レジスタからクロック同期して出力される。Blitクロックは、これと同様に他の機能も制御している。

(以下余白)

データの第1平面が5ビット・レジスタに記録されると(プロセッサは一回に1平面のみを操作する。)、次の水平に隣接したバッチからの新しいデータ平面がバレル・シフタにより処理され、5ビット・レジスタに記録される。各811クロックの循環の最後に、5つの4ビット2:1マルチプレクサ(MUX)622,624,626,628,630の入力端子に8ビットデータが転送される。各MUXは、各バッチ平面の5列データのの一つを処理する。明確化のため、5ビット・レジスタに既に保存されたデータを前バッチ平面、レジスタの入力端子にあるデータを現バッチ平面と呼ぶことにする。各マルチプレクサは、前バッチ平面から1列を、現バッチ平面から対応する列を受信する。Xシフトコントロールバス610のデータ制御に従って、2:1 MUX群は、前バッチ平面と現バッチ平面とをマージする。コントロールバスの各ビット610(0), 610(1), 610(2), 610(3), 610(4)が、直接一つのMUXを制御する。

X方向のマージ操作を例示する。以下に示す例

上記の例では、5つの4ビット2:1マルチプレクサの2つ(本例では622,624)が、前バッチ平面の最初の2列を選択し、それらを残りの3つのマルチプレクサ626,628,630により選択された現バッチ平面の最後の3列とマージしている。一度設定されると、マルチプレクサ・プログラミングはすべてのラスタ(ブロック・コピー)操作を確実に実行する。

デコーダPAL 632のプログラミングについて説明する。上記の例から、バレル・シフタ602,604, 606,608,610は、MUX 622,624,626,628,630と相互に関連して処理を実行していることがわかる。例えば、前バッチ平面がNポジション左にシフトしたとすれば、MUXは前(保存された)バッチ平面の最初の5-N列と、バレル・シフタによりシフトされた現(保存されていない)バッチ平面の最後のN列を選択しなければならない。

この方式は、逆の場合も同様である。前バッチ平面の最初の5-N列と、現バッチ平面の最後のN列の選択は、5ビットXシフト制御データが5

の1-1の2つの隣接する画素データのバッチを、3ポジション左にシフトすると仮定する。

例1-1

第1バッチ

A15-A16-A17-A18-A19

A10-A11-A12-A13-A14

A05-A06-A07-A08-A09

A00-A01-A02-A03-A04

第2バッチ

B15-B16-B17-B18-B19

B10-B11-B12-B13-B14

B05-B06-B07-B08-B09

B00-B01-B02-B03-B04

バレル・シフターを通過した後のバッチは、以下のようなになる:

前バッチ

A18-A19-A15-A16-A17

A13-A14-A10-A11-A12

A08-A09-A05-A06-A07

A03-A04-A00-A01-A02

現バッチ

B18-B19-B15-B16-B17

B13-B14-B10-B11-B12

B08-B09-B05-B06-B07

B03-B04-B00-B01-B02

Xシフトおよびマージ論理回路は、以下に示すような新しいバッチを作成する:

A18-A19-B15-B16-B17

A13-A14-B10-B11-B12

A08-A09-B05-B06-B07

A03-A04-B00-B01-B02

つの2:1 MUX 622,624,626,628,630の入力端子を制御することにより実行される。デコーダPAL 632は、3ビットのバレル・シフタ制御データが順番にNの循環シフトを実行するようプログラムされている。Xシフト・レジスタ制御ライン610上の論理値1の宣言(assert)によって各MUXの現入力端子が選択されるならば、デコーダPAL 623は、バレル・シフタにNと同数分のシフトをさせるために、単にXシフト・レジスタ制御ライン610の1の数を3ビットの2進数に変換すれば良いのである。

本実施例では、Yシフトが上下に移動するのに対し、Xシフトは左移動のみとなっている。Yの上下移動は、ソースとデスティネーションの四角形を重ね合わせるために必要である。すなわち、デスティネーションがソースのボトムと重なっている場合は、デスティネーションに転送する前にソースのトップがボトムに来るようにコピーしなければならない。X方向の場合は、Bit RAMにより問題は生じない。これは、行が常にデスティネー

ションに書き込まれる前に処理されるからである。本実施例ではXシフトを左移動のみとしたが、右移動も簡単に実行できるということを理解されたい。Xシフトは循環しているため、Nの右シフトは、 $5-N$ の左シフトと全く同じことだからである。

上記説明から、Blitクロックの各循環の間に、Xシフトおよびマージ論理回路512から新しくXシフトされた平面のデータが出力されることがわかる。

g) 時間領域多重化方式

Xシフトおよびマージ論理回路は、時分割多重方式(TDM)を使用することにより、一度に一つ以上の平面を処理できるように変更されている。本実施例のXシフトおよびマージ論理回路は、第12図に示すようにTDMおよび非TDMの双方に適合する。多くの場合、TDM操作は、Blitクロックの速度を2倍にし、各プロセッサ・クロック・サイクルの各バッチからの2平面を処理する。入力バ

ーズ画素平面の下および第2フェーズ画素平面の上に宣言されるフェーズ1信号1006を供給する。言い換えれば、Xシフトおよびマージ回路には2つの完全な記憶レジスタのセットがあるということである。各セットはデュアル・フェーズ・バッチ・サイクルの1フェーズのみ、1平面のみに使用される。

第12図から、第1セットのレジスタ612, 614, 616, 618の全クロック入力端子は第1ゲート1216の出力端子と、第2セットのレジスタ1202, 1204, 1206, 1208の全クロック入力端子は第2ゲート1214の出力端子と接続していることがわかる。TDM操作の第1フェーズ実行中は、ゲートは第1セットのレジスタのみにデータをクロックする(フェーズ1信号はロー)。TDM操作の第2フェーズ実行中は、ゲートは第2セットのレジスタのみにデータをクロックする(フェーズ1信号はハイ)。同様に、インバータ1210, 1217はフェーズ1信号がローのときに第1セットのレジスタを、フェーズ1信号がハイのときに第2セットのレジ

スタ・レジスタ312は、最初のBlitクロック・サイクルでクロック同期して入力されたバッチの1平面が選択され、次のBlitクロック・サイクルで別の平面が選択されることを除けば、通常通り作動する。TDM操作では、Blitクロック1004(入力バッチ・レジスタのクロックアウトを行うのに用いられるクロック)は、プロセッサ・クロック1000(すなわち入力バッチ・レジスタ312に完全なバッチをクロック同期して入力するクロック)の2倍の速さで作動することに注意しなければならない。同様に、出力バッチ・レジスタ330は、各クロック・サイクルに、画像メモリに書き込むためのバッチ平面を、一面ではなく2面ロードするよう調整されている。

第12図のXシフトおよびマージ論理回路は、第2のセットである4つのレジスタ1202, 1204, 1206, 1208およびゲート回路1210, 1212, 1214, 1216が追加されたことを除けば、第6図示の実施例とほぼ同様である。さらに、グラフィックス・プロセッサ308は、処理されたバッチの第1フェ

スタを出力可能とする。TDM操作は、2つのフェーズが存在していると概念化できる。第1フェーズはバッチ入力レジスタ312に第1バッチがクロックされたときに開始する。第1フェーズでは、バッチの第1平面が選択され処理される。第1平面は、Xシフトおよびマージ論理回路に到達すると、シングル平面の例の場合と同様に処理される。グラフィックス・プロセッサはフェーズ1信号をローに宣言しているため、シフトされたデータは第1セットの5ビット・レジスタ612, 614, 616, 618にクロック同期して記録される。次に、グラフィックス・プロセッサ100がフェーズ1信号の宣言を解除(ハイに設定)し、バッチ入力レジスタ312の第2バッチ平面が選択される。(第1バッチの)第2平面は、第2セットのレジスタ1202, 1204, 1206, 1208にシフトされロードされる。既に記述したようにフェーズ1信号の宣言(ロー)中は、第1セットの5ビット・レジスタはロードされ、出力可能となるが、第2セットではこれは起こらない。宣言解除(ハイ)中は、第

2セットの5ビット・レジスタはロードされ、出力可能となるが、第1セットではこれは起こらない。

第1バッチの第2バッチ平面が第2セットの5ビット・レジスタにクロックされると、第2バッチが入力バッチ・レジスタ312にロードされる。フェーズ1^{*}信号が再びローに宣言されると、入力バッチ・レジスタは、最初のフェーズ1^{*}信号がローであるときにクロックされたものと同じバッチ平面を最初に選択する。例えば、第1バッチのバッチ平面1が最初に選択されたとすれば、同様に第2バッチのバッチ平面1が最初に選択される。(第2バッチの)選択されたバッチ平面は、パレル・シフタによりシフトされ、レジスタの入力端子および2:1 MUXに転送される。フェーズ1^{*}が宣言(ロー)されているため、第1セットのレジスタのみが出力可能となる。各Xシフトコントロールバス610の1ビット制御により、MUXは第1および第2バッチの第1バッチ平面のマージを実行する。次の811tクロック(第2フェーズの開始

バッチに対し、2サイクルで進行している。このバッチ・データは参照番号1002で表されている。811tクロックとバッチ・データの関連タイミングにより、クロックインされた各バッチにつき2つのバッチ平面が入力バッチ・レジスタ308(参照番号1008で表されている)からクロックアウトされる。フェーズ1^{*}信号は、各バッチがクロックインされるごとに1回循環し、各バッチの第1バッチ平面をXシフトおよびマージ回路の第1セットのレジスタにクロックインさせ、第2バッチ平面を第2セットのレジスタにクロックインさせる。一連のバッチの処理過程は、参照番号1010で表されている。

非TDM(第11図)では、第12図の回路の動作は少し異なる。非TDM操作(シングル・フェーズ)では、811tクロック1004は、入力レジスタに記録された各バッチに対し1サイクルで進行し、フェーズ1^{*}信号1006は永久的にローの状態になっている。このタイミングでは、各バッチにつき1つのバッチ平面のみが入力レジスタからクロックアウト

をマークする)で、現行の(保存されていない)データが第1セットのビット・レジスタ612,614,616,618にロードされる。

同じプロセッサ・クロック・サイクルの第2フェーズでは、(第2バッチの)第2バッチ平面(すなわち、第1バッチで2番目に選択された平面)がバッチ入力レジスタにより選択される。グラフィック・プロセッサはフェーズ1^{*}信号を解除(ハイに設定)し、第2セットの5ビット・レジスタが出力可能となる。2:1 MUXは、第1および第2バッチの第2平面のマージを実行する。次の811tクロックで、第2バッチの第2平面は第2セットの5ビット・レジスタにクロックされる。このサイクルは、全行のバッチすべての処理が終了するまで継続する。

第12図の回路を使用したTDM操作のタイミングは、第10図を参照すると理解しやすい。第10図から明らかなように、TDM操作では、811tクロック1004は、可逆データ・バス306から入力レジスタ312にクロック同期して記録されたデータの各バ

ットされる(参照番号1008'で表されている)。その結果、バッチ平面の行は、参照番号1010'に示すように一般的な順序で処理される。第12図の回路においてフェーズ1^{*}信号が永久的にローである状態は、第6図の回路と同じ動作となる。

Xシフトおよびマージ論理回路は、本例の5×4とは異なる他のバッチ・フォーマット用にも簡単に修正できる。パレル・シフタ602,604,606,608,610は、各バッチ列につき1ビットを有していなければならない。これは、レジスタでも同じである。各バッチ列には一つの2:1 MUXが割り当てられ、MUXは、バッチ行の数だけ設けられなければならない。本例では、Xシフトおよびマージ論理回路には、プログラマブル論理アレー(PAL)が使用されている。

本書には、「隣接する」という用語が使用されている。この用語の意味は、2次元の指定可能な画像メモリ302の環境を考えると理解しやすい。このメモリはビデオ画面と同じようにアドレスリングされているため、データは列と行との配列に

格納されていると考えて良い。従って、与えられたパッチは、垂直方向 (Y) に隣接する2つのパッチ (すなわちパッチ行の上下にあるもの) および水平方向 (X) に隣接する2つのパッチ (1列の両側にあるもの) を有している。

h) Yシフト

Yシフト論理回路316は、第7図を参照すると理解しやすい。Yシフト論理回路512は、Xシフトおよびマージ論理回路314から、Xシフトされたパッチ平面を受信する。Yシフト回路内では、Yシフトコントロールライン712の2ビット制御データに基づいて、パッチ平面の行を上下にシフトするため、5つのビット・バレル・シフタ702, 704, 706, 708, 710が使用される。各バレル・シフタには、パッチ平面の完全な1つの列 (4ビット) がロードされる。

Xシフトされたパッチ平面の0, 5, 10ポジションのデータから成る第1列は、第1バレル・シフタ702にロードされる。ポジション1, 6, 11, 16の

たパッチ平面は、Yシフト回路316の出力端子から、ライン・ストレージRAM318に転送され、Yマージおよび行の集積が実行される。

i) ライン・ストレージRAM

ライン・ストレージRAMは、2つの目的に用いられる。第1に、完全な出力パッチ平面を作成するために、現および前入力パッチ平面から選択された行をマージする。第2に、ページモードで画像メモリ302に直接読み込ませることができるように、完全にシフトおよびマージされたパッチ平面のすべての行を格納する。

ライン・ストレージRAMの動作は、第8図を参照すると理解しやすい。第8図は、アドレス・ジェネレータ333およびライン・ストレージRAM318の双方を示している。アドレス・ジェネレータ333は、10ビット・カウンタ802、プログラマブル論理アレー (PAL) 804およびトライステートバッファ806を備えている。

ライン・ストレージRAM318は、書き込み可能な

データから成る第2列は、第2バレル・シフタ704にロードされる。ポジション2, 7, 12, 17データから成る第3列は、第3バレル・シフタ706にロードされる。ポジション3, 8, 13, 18のデータから成る第4列は、第4バレル・シフタ708にロードされる。ポジション4, 9, 14, 19のから成る第5列は、第5バレル・シフタ710にロードされる。

2本のYシフトコントロールライン712は、グラフィックス・プロセッサ308により制御されている。2ビットのYシフト情報は、Yバレル・シフタ702, 704, 706, 708, 710に各パッチ平面の4行をY方向に順番に3ポジションまでシフトさせるのに充分である。Y方向への4ポジションのシフトは、単に5×4パッチ平面を開始位置に戻すことにすぎないため不要である。シフトは循環しているため、負のシフトも簡単である。すなわち、1ポジションの負のシフトは、3ポジションの正のシフトと全く同じだからである。

Yバレル・シフタにより処理されたYシフトレ

論理回路808および4つの2K×5ランダム・アクセス・メモリ810, 812, 814, 816 (Bit RAM) を具えている。グラフィックス・プロセッサ308は、いくつかのデータを供給し、アドレス・ジェネレータ333およびライン・ストレージRAM318へのラインを制御している。これらのラインには、1本のダイアグノスティック・リード・イネーブルライン818, 10本のプロセッサ・データ・ライン820, 1本のカウンタ・イネーブルライン822, 1本のカウンタ、ロード・イネーブルライン824, blitクロック・ライン826, Blitリード・ライン826, Blitライト・ライン828, 2本のYシフト制御ライン712, ダウン制御ライン832, および偶数制御ライン834が含まれている。

ライン・ストレージRAM318には、2つのモードがある。パッチ平面の完全な画像の行がBlit RAMから読み込まれるBlitリード・モード、およびパッチ平面の完全な画像の行がBlit RAMに書き込まれるBlit RAMライト・モードである。

まず、Blitライト・モードについて説明する。

パッチ平面の完全な行は、継続的にXYシフト後のデータ・バス322に送信される。XYシフト後のパッチ平面は、Blitクロック・サイクルごとに、Yシフト論理回路からバス322へ一つずつ送信される。パッチのラインを処理する前に、グラフィックス・プロセッサは、プロセッサ・データ・バス820に初期アドレスを設定し、ライン824にロードイネーブル信号を宣言することにより、アドレス初期値（通常0）で、10ビット・カウンタ802をロードする。初期値がロードされ、最初の有効なパッチ平面がXYシフト後のデータ・バス322に送信されると、ロードイネーブル信号は解除され、ライン872にカウントイネーブル信号が宣言される。10ビット・カウンタのデータは、ライン820上のBlitクロック1004のサイクルごとに増加される。カウンタ・データは、blit RAM 810, 812, 814, 816の下位10桁のアドレスビットのアドレスとして使用される。トライステートバッファ806は診断の目的で使用され、グラフィックス・プロセッサ308がリードイネーブル*信号

い。

Blit RAMs リード・モードも同様である。10ビット・カウンタ802は、グラフィックス・プロセッサ308によりロードされ、カウント可能となる。次に、ライン826のBlitリード*信号が宣言（ロー、真）され、Blit RAMsの出力をイネーブルとする。（ライン828の）Blitライト*信号は解除（ハイ）されているため、Blit RAMsの書き込みは禁止されている。Blit RAMから完全な出力パッチ平面を読み込むために、PAL 804のデータ制御により、すべてのBlit RAMの上位アドレス・ビットがセットあるいはリセットされる。

各Blit RAMには、パッチ平面の1行が割り当てられていることに注意しなければならない。行1のRAM810は、XYシフト後のパッチ平面の第1行、すなわちXYシフト後のデータ・バスの0から4ビットのみの受信または書き込みを実行する。行2のRAM812は、XYシフト後のパッチ平面の第2行、すなわち5から9ビットのみの、行3のRAM814は、XYシフト後のパッチ平面の第3

（ローが真）を宣言することによりカウンタ・アドレス・データを読み戻すことができる。この信号を宣言すると、カウンタ・データはプロセッサ・データ・バスライン820上に送信される。

Blitライン・モードを開始する前に、Blit書き込み信号が、ライトイネーブル論理の入力ライン828がローに宣言される。これにより、blitクロック1004は、XYシフト後のデータ・バス322から送信される有効パッチ平面データをRAM810, 812, 814, 816に書き込み可能とする。Blitライト・サイクル中、PAL 804は、Blit RAMに完全な出力パッチ平面を正確に作成するべく、グラフィックス・プロセッサからの制御データに従って、各Blit RAMの上位アドレス・ビットをセットあるいはリセットする。

時間領域多重化方式では、Blit RAMカウンタ802は、シングル・フェーズ（非時間領域多重化方式）の2倍の速さでクロックされ、Blit RAMは、各パッチサイクルごとに（1つではなく）2つのパッチ平面を格納することに注意された

行、すなわち10から14ビットのみの、行4のRAM816は、XYシフト後のパッチ平面の第4行、すなわち15から19のみの受信または書き込みを実行する。

PAL 804は、Blitリード・モードおよびBlitライト・モードで、Blit RAMを適切に使用可能にするために使用される。第8図から明らかなように、PAL 804は、各Blit RAMに上位アドレス・ビットを供給する。各RAMは、実際に、2つの分離したアドレス指定可能領域を持っていると考えることができる。第1領域は、PALがBlit RAMの上位アドレス・ビットを設定したときにアドレス指定され、第2領域には、PALが下位アドレス・ビットを設定した時にアドレス指定される。これらの領域を、それぞれ、第1および第2領域と呼ぶ。各Blit RAMは、パッチ平面データの1行（5ビット）を格納する。1アドレス領域は、完全なパッチを保存するのに使用され、他のアドレス領域は、開始されたパッチの保存に使用される。

5×4パッチの何行かが与えられた数値Nによ

りYシフトされ、Y方向に隣接するパッチとマージされると仮定する。ライン・ストレージRAMは、この処理をいくつかの段階で行う。

Yマージに必要な正確なアドレス処理は、Yシフトの量と、処理される四角形領域の垂直移動方向とによる。代表的なYシフトおよびマージ操作は、デスティネーション画像領域がソース画像領域より上に位置しているようなソース領域の処理例で説明することができる。ラスト操作を正確に実行するために、ソース領域はトップダウン処理されなければならない。この理由は、ソースとデスティネーションの画像領域が重複することを考えると理解しやすい。重複させる領域へのオーバーラップを防ぐために、トップダウン処理が実行される。

例えば、パッチ内のデスティネーション画素がソース画素よりN画素上方にあるとする。シフトアップする場合、PAL804は、パッチの行1を最下行、行4を最上行とみなす。

以下に、これについての一般的な処理方法につ

ドレス領域に保存され、N+1から4までの行は関連Blit RAMの第1アドレス領域に保存される。この段階では、Blit RAMの第1アドレス領域には、パッチ平面の完全な1行が含まれている。Blit RAMの第2アドレス領域には、コピーされるパッチ平面の次ラインの最初のN行が含まれている。

次の段階で、パッチ平面が読み込まれる。Blit RAMが読み込み可能となり、すべてのBlit RAMの第1領域のパッチ平面データが読み込まれる。既述したように、この時点では、第1アドレス領域には完全なパッチの表示行が含まれている。従って、グラフィックス・プロセッサ308はページ・モードですべてのパッチ行を読み込むことができる。

次の段階で、もう一つのパッチを作成する。この段階で、PAL804は再度Blit RAMの上位アドレスを逆にする。Yシフトされたパッチ平面の新表示行はYシフト回路のライン・ストレージRAMから送信される。行1からNは、関連Blit RAMの第

いて述べる。

第1段階は準備段階である。(Yシフト論理回路316によって)Y方向へN行シフトアップされたパッチ平面の第1ラインが、ライン・ストレージRAM318に送信される。最初のソース画像ラインの前パッチに対し、シフトされた1からN行が関連Blit RAMの第1アドレス領域にロードされる。N+1から4までの行は、関連Blit RAMの第2アドレス領域にロードされる。この段階の終了では、Blit RAMの第2アドレス領域には、出力パッチ作成に不要でBlit RAMに読み込まれないパッチ平面の行が含まれ、第1アドレス領域には、画像メモリに出力される可能性のあるパッチ平面の第1行または複数の行が含まれている。

次の段階で、パッチを作成する。この段階で、PAL804は、Blit RAMの上位アドレスを逆にする。Yシフトされたパッチ平面の第2ラインは、Yシフト論理回路316のライン・ストレージRAMから送信される。この時、第2ソース画像ラインのシフトされた行1からNは、関連Blit RAMの第2ア

ドレス領域に保存される。N+1から4までのパッチ平面行は、関連Blit RAMの第2アドレス領域に保存される。この段階では、Blit RAMの第2アドレス領域には、パッチ平面の完全な1行が含まれている。コピーされるパッチ平面の次ラインの最初のN行は、関連Blit RAMの第1アドレス領域に保存される。

次の段階で、もう一つのパッチ平面の読み込みを行う。Blit RAMが読み込み可能となり、すべてのBlit RAMの第2アドレス領域のパッチ平面データが読み込まれる。既述したように、この時点で第2アドレス領域には、完全なパッチの全行が含まれており、グラフィックス・プロセッサ508は、行アドレス・モードで(パッチ行の)前ラインを読み込むことができる。

パッチ平面の作成と読み込みは、すべてのデータ・ブロックのコピーが終了するまで、交互に実行される。N値のYシフトを実行するアルゴリズムの概要をまとめると、以下のようになる。

A. 準備段階 - 各パッチ平面の最初のN行を、

Blit RAMの第1アドレス領域に書き込む。各バッチ平面の次の4-N行を、Blit RAMsの第2アドレス領域に書き込む。

B. 第1バッチ平面の作成段階—各バッチ平面の最初のN行を、Blit RAMの第2アドレス領域に書き込む。各バッチ平面の次の4-N行を、Blit RAMの第1アドレス領域に書き込む。

C. 第1バッチ平面の読み込み段階—Blit RAMの第1アドレス領域を、ページ・モードで読み込む。

D. 第2バッチ平面の作成段階—各バッチ平面の最初のN行を、Blit RAMsの第1アドレス領域に書き込む。各バッチ平面の次の4-N行を、Blit RAMの第2アドレス領域に書き込む。

E. 第2バッチ平面の読み込み段階—Blit RAMの第2アドレス領域を、ページ・モードで読み込む。

F. コピーするデータを持つバッチ行の総数だけ、BからEまでの段階を繰り返す。

ている。これらには、Yシフト制御ライン(2ビット)、ダウン・制御ライン、偶数・制御ライン、およびBlitソード・制御ラインが含まれている。Yシフト制御ライン112は、Yシフトの量(本例ではN)をキャリーするのに用いられる。これらのラインは、Yシフト論理回路316により、同じYシフト信号を処理する。グラフィックス・プロセッサは、バッチ平面のすべての偶数行に偶数ライン・(ロー)を、すべての奇数行にハイを宣言する。このように、PAL 804はどの段階が実行されているのかを常に把握している。また、PAL 804は、現行モード(ライトまたはリード)を決定するために、Blitリード信号を使用する。ダウン・制御ライン832は、バッチ行の読み込みがどの方向(上または下)に処理されているかという情報を、PAL 804に伝達する。このラインのロー信号は、処理が画像メモリの上から下へ実行されていることを示すのに用いられ、“1”信号は、下から上へ実行されていることを示すのに用いられる。リード・モードでは、PAL 804は

第1ラインが、(最初の出力行のマスクされた書き込みも含めて)最初の出力ラインに必要なすべての行を含んでいる場合には、準備段階は不要であることに注意されたい。

上記のアルゴリズムは、N行シフト・ダウンする場合にも適用されるが、この場合、バッチの処理は、下から上に実行される。デスティネーション画像領域が、ソース画像領域の下に位置している場合には、PAL 804は、各バッチ平面の最上行を第1行、最下行を第4行とみなす。従って、読者は、この第2の番号規則に従うものとする。ソースとデスティネーションのバッチをオーバーラップした時に、コピー結果が無効データとなる危険がある場合には、シフト・ダウンおよび下方への処理(またはその逆)を実行してはならない。

PAL 804は、各段階で、Blit RAMの上位アドレス・ビットを正確に作成するようにプログラムされている。このため、グラフィックス・プロセッサ308からいくつかの制御およびラインを利用し

すべてのRAMについて同じアドレス領域(すなわち、最新の出力バッチ行が作成された領域)を選択する。ライト・モードでは、完全な出力バッチを作成するために、種々の入力を使用して、適切な領域の正しい行のマージが実行される。

J) 論理装置

論理ユニット317は、できれば演算論理ユニット(Arithmetic logic Unit:ALU)が望ましい。本例に係る論理ユニットは、Texas instruments製の5つの74AS181 ALUを備えている。このALUの目的は、ソースとデスティネーション平面との間で、ブール演算を実行することである。

論理演算を実行する場合には、出力バッチがライン・ストレージRAM 318から読み出される前に、デスティネーション平面が入力バッチ・レジスタに読み込まなければならない。デスティネーション・バッチは、ソース・バッチおよびソースとデスティネーション平面がALU 318の入力端子に同時に送信された後に、直接読み込むこ

とができる。これは、画像メモリ302へ行を書き込み中はページ・モードオペレーションができないことを意味しているが、デスティネーションの読み込みはページ・モードで行うことができる。

(本例の画像メモリ302のような)ビデオRAMへ書き込む場合には、デスティネーション・パッチの読み込みは、RAMの内部論理モードを使用する事により、回避することができる。論理モードを設定すると、日立社のVRAMは、通常の書き込みサイクル時に、入力データとRAMデータの間で論理演算を実行する。

ビデオRAMの操作についての概要は、HITACHI ICMEMORY DATA BOOK, 1986 version (日本の日立社および米国のHITACHI販売店にて入手できる)に説明されている本内容には、そのすべてが参考として統合されている。

k) 出力マルチプレクサ

出力マルチプレクサは、基本的には、グラフィ

1302は、入力端子の制御ラインの状態に従って、処理されたパッチ平面を格納する8つの出力レジスタ1308, 1310, 1312, 1314, 1316, 1318, 1320, 1322のうち一つを選択する。

出力レジスタPALの制御ラインには以下のものが含まれる。時間領域多重化処理のフェーズ1に与えられた平面のレジスタをPAL 1302に選択させる3本のフェーズ1出力平面選択ライン1304、時間領域多重化されたblit処理のフェーズ2に与えられた平面のレジスタをPAL 1302に選択させる3本のフェーズ2出力平面選択ライン1306、実行中の操作がシングル・フェーズであるかデュアル・フェーズであるかをPAL 1302に伝達するデュアル・ライン930 (ロー、真)、フェーズ1およびフェーズ2の平面選択入力端子を正確に選択させるために時間領域多重化処理の現行フェーズをPAL 1302に伝達するフェーズ1ライン932、およびBlitプロセッサが画像メモリからデータを読み込んでいる時、出力レジスタへのすべてのロードを禁止するために使用するレジスタ・ラ

ックス・プロセッサ308により提供される選択入力端子を持つ20ビット2:1MUXである。出力マルチプレクサにより、外部(例えば、他の同期Blitプロセッサ)から供給されたパッチの平面を、ソース・パッチの平面のいかなる位置にも挿入することが可能である。外部から供給された平面は、ソース平面のXおよび/またはYシフトが実行されているいかににかかわらず、挿入することができる。外部から供給される平面とソース平面の選択は、グラフィックス・プロセッサ308により制御されることを、理解されたい。

1) 出力パッチ・レジスタ

出力パッチ・レジスタ330 (第13図)は、入力パッチ・レジスタ312と同様の処理を実行する。XYシフト・データの選択平面(20ビット)を160ビット・パッチの適切な位置へ入力するため、グラフィックス・プロセッサ308により生成されるいくつかの信号が、出力レジスタ論理回路PAL 1301の制御に使用される。この場合、PAL

イト・イネーブルライン1324である。

PAL 1302は、データを適切な平面レジスタにロードするために、(グラフィックス・プロセッサ308により生成される)Blitクロック1004を使用して、正確な時間にレジスタ・クロック入力にライジング・エッジを作成する。シングル・フェーズ操作では、1パッチ・サイクルにつき1つのライジング・エッジのみが作成される。2フェーズ操作では、2つのライジング・エッジが作成される。PALへの書き込み可能入力、Blitプロセッサが画像メモリからデータを読み込んでいる時、出力レジスタへのすべてのロードを禁止するために使用される。

出力レジスタは、画像メモリ・データ・バス306の160ビット上のパッチデータの全8平面を使用可能とすることを理解されたい。しかしながら、シングル・フェーズ走査では1平面のみが、2フェーズ走査では2平面が有効となる。有効な平面のみが、グラフィックス・プロセッサ308によって、画像メモリに書き込み可能となる。

画像メモリ平面イネーブル機能は、できれば画像メモリ308自身により実行されるのが望ましい。本例の画像メモリ内の各ビデオRAMは、4平面を格納する。しかしながらビデオRAMは内部書き込みイネーブル機能を備えており、それによりすべての平面の書き込みを禁止することができる。グラフィックス・プロセッサ308内の平面保護レジスタの内容は、書き込みサイクルおよびVRAMの内部書き込み禁止中のチップのデータラインに表れる。VRAMを使用しない他のタイプの画像メモリでは、画像の各平面は、しばしば別のRAMチップのセットを使用して格納される。その場合、書き込み信号により、ある平面のサブセットは書き込み可能となる。各平面への書き込み信号は、8ビット平面使用可能レジスタの関連ビットのANDゲートが有効に使用される。

入力レジスタ平面選択926,928 (第9図)は、出力レジスタ平面選択1304,1306 (第13図)とは独立している。このため、平面間のコピー(すなわち、1平面のソースから他の平面のデスティ

ネーションの一部は、最終的に画像1404のように表されると仮定する。マスクしない状態ではBlitプロセッサはソース画像領域1402全体を、デスティネーション画像1404上にオーバーレイする。これでは、希望の結果にならないことは明白である。

書き込みマスクは、また、別の目的にも使用される。最上行および最下行のバッチ並びに、各行の左右バッチは、デスティネーションの境界が正確なバッチ境界となっていない場合、マスクが必要となる。

ソース画像の要求部分のみを正確にコピーするためには、ソースの一部をコピーさせないようにする必要がある。各画素のコピー情報のマップは、ソース画像の1平面(マスク平面と呼ぶ)に格納されている。(各ソース・バッチ内の)この平面は、バッファ320を経由して書き込みマスク・レジスタにロードされ、各バッチ平面の形の輪郭に従って、領域のマスキングが実行される。このような操作はマスク・コピーと呼ばれ、所望の

ネーションへのコピー)が可能となっている。これは、マルチビット画素のシフト操作、または多くの平面を持つフレームストアの平面間での1(またはそれ以上の)ビット画像のコピーに有効である(例えば、8ビットフレームストアに大型の1ビット画像を格納する場合、有効である)。平面間コピーを実行する場合、プロセッサは、処理した平面を出力レジスタ論理回路内の選択平面レジスタに別々に入力するため、入力および出力バッチ・レジスタのソースおよびデスティネーション平面の制御は別々に行われる。

m) 書き込みマスク

書き込み(ライト)マスク・レジスタ320は、デスティネーション領域のマスクを生成するために、グラフィックス・プロセッサ100により使用される。マスク・レジスタの機能は、第14図に最も良く表されている。ソース領域1402が、デスティネーション領域1404にコピーされると仮定する。さらに、ソースよりコピーされるデスティ

ネーション画像1106が生成される。

マスク・コピーが時間領域多重化方式を使用するシステムのみで実行可能であることを、理解されたい。各バッチのマスク・データは、画像データ・バッチが出力バッチ・レジスタにロードされる時に、書き込みマスク・レジスタにロードされなければならないからである。これにより、各出力バッチからの画素の書き込みは、Blitプロセッサを経由した画像データと同じデスティネーションポジションに正確にシフトおよびマージされた該当ソース・マスクにより、修正される。マスク・コピーは、マスク平面に従って処理される1画像平面のみが必要であり、多重平面画像の処理はバスの数を増加すれば良いということがわかるだろう。

マスク・コピーのメカニズムは、システムごとに異なっている。本実施例では、各画素ポジションは、画像メモリ内の他の画素ポジションと重複しないRAMバンクを使用している。本例の画像メモリには20の画素ポジションがあり、5×4のバ

パッチ形式の各ポジションに対応している(第4図参照)。従って、パッチ画素の選択されたサブセットのみを書き込む場合は、使用可能なRAM信号の一つ(好ましくは列アドレス・ストロブ)がゲートされ、ディスエーブルとなっているRAMバンクには到達しない。グラフィックス・プロセッサ308は、20ビット書き込みマスク・レジスタの各ビットを使用して、1本の列アドレス・ストロブ(CAS)ラインをゲートする。本例では、PALのゲーティングは、書き込みマスクビットと列アドレス・ストロブとを効果的に“AND”処理することにより、実行される。これは一例に過ぎず、グラフィックス・プロセッサ内では、他にも様々なバッファリングおよび論理が使用される。本例のグラフィックス・プロセッサでは、書き込みマスクにこの方法が使用されているが、他の方法を用いても同じ結果が得られると予想されることを理解されたい。

本発明には、2つの異なるパッチ・マスク操作があることが理解されたい。第1の操作は、境界

n) ページ・モード・アドレス指定

多くの動的記憶装置のように、画像メモリ502は、行アドレス・ストロブ(Row Address Strobe:RAS)がステابل状態でよく、列アドレス・ストロブ(CAS)のみを循環させるページ・アクセス・モードを備えている。このモードの操作では、各行アドレスのアクセスを、240nsから約120nsに短縮する。Blitプロセッサは、各操作サイクルのデータのすべてのアドレス行を格納しているため、ページ・モード操作が可能である。従って、処理したデータの全行をRAMアドレスを変更することなしに読み込むことができる。ページ・モード操作は、従来の技術でも良く知られている。Blitアーキテクチャは、プログラマまたはシステム・デザイナーがより速くラスタ操作を実行できるよう、このモードの使用を可能にしている。

o) シフト計算

既述したように、シフトおよびマージ論理回路

がパッチと適合しない場合、デスティネーションの最上、最下行をマスクすることである。第2の操作は、マスク・コピーを実行して、シフト・マージ論理回路にマスク平面を転送することである。マスク・コピーの最上行を実行する場合、エッジマスクは、データ・マスクと共に“AND”処理が実行される。これはグラフィックス・プロセッサ308で実行され、Blitプロセッサからのマスク・データが読み込まれ、現行エッジマスクと“AND”処理された後、結果が書き込まれマスク・レジスタに書き込まれる。

マスク論理演算中、Blit論理ユニット317は、フェーズごとに異なる操作を実行する必要がある。論理ユニットは、画像データに要求された論理演算を実行するが、マスク・データをストレートにパスしなければならない。このため、論理回路317には、グラフィックス・プロセッサにより、別々のフェーズ1命令およびフェーズ2命令が供給される。命令(OPコード)は、フェーズ1信号により選択される。

は、XおよびY方向に与えられた数値(すなわち0以上、パッチ寸法(dimensions)以下のすべての数値)に従って、シフトおよびマージを実行する。本例の5×4パッチ寸法の場合、Xシフト数は、デスティネーション・パッチ内の所望の画素のXポジションからソース・パッチ内の画素のXポジションを差し引き、結果が負数ならば5を加算することにより計算される。Yシフトは、パッチが読み込まれる方向により異なる。ソース領域が上から下へ読み込まれる場合は、ソース領域のパッチ内の画素ポジションからデスティネーション領域内の画素ポジションが差し引かれる。結果が負数ならば、4を加算する。次に、Yシフト数を得るために、4からその結果を差し引く。コピーが下から上へ実行される場合は、デスティネーションのポジションからソースの画素ポジションが差し引かれ、結果が負数ならば4を加算される。

(5×4以外の)その他のパッチ寸法の場合でも、パッチ寸法に影響を及ぼす定数の変更がない

限り、計算方法は同じである。つまり、X方向では、5が加減される代わりに、パッチのX寸法が加減される。Y方向では、4が加減される代わりに、パッチのY寸法の値が使用される。言い換えれば、X方向にH画素、Y方向にV画素を持つパッチのシフト値を計算する場合は、5および4にそれぞれHおよびYを代用すればよい。

III. 平面スワッピングおよびビット・ポジション処理

前章では、XおよびY方向のシフトおよびマージ機能について説明した。パッチ内の各画素の交換、おきかえ、およびパッチ周辺への移動は、マージ機能を使用しなくても、シフト機能だけで実行できる。すなわち、パッチが3次元の配列であるとすれば（本例ではパッチは、 $5 \times 4 \times 8$ の3次元である）、本システムおよび方法は、パッチ内のすべての次元（X、YまたはZ）のすべてのビットに関し、おきかえまたは移動を実行することができる。

Blit RAMの書き込みが初期化されるまではそれを画像メモリに書き込まないためである。さらに、垂直隣接マージとは別に、パッチ内の全平面の全行数を他のラインのデータと置換することもできる。

Xシフトおよびマージ論理回路314には、多少の変更を加えることができる。通常、デコーダPAL 632は、MUXの選択入力に対応するバレル・シフタ602, 604, 606, 608にシフト制御信号が送信されていることを確認するために使用される。しかし、望むのであれば、デコーダPALを取り除き、バレル・シフタに送信されるシフト制御データをMUXの選択データと切り放して処理させることができる。簡単な例としては、マージなしのシフトを実行することにより、パッチ1平面または複数の平面を、数値に従ってXシフトさせることができる。

マージなしのXおよびYシフトとは、全ビットを平面内のどのポジションにでも移動させることができ、しかも平面を回転させることができること

a) ビット・ポジション処理

XおよびYシフト論理は、与えられた平面の周囲にビットを移動することができる（すなわち、平面処理）。ビット列の上下のシフトについては、Yシフト論理316の現形式で実行できる。つまり、パッチの与えられた平面において、画素対画素で選択した行数の上下移動が可能である。これは、一つのパッチ内の20画素をそれぞれ1ビット・ポジションだけシフトすることと同じである。Yシフト論理は一度に1平面ずつ処理を実行するため、他をシフトせずにいくつかの平面のみをシフトさせることができる。

この処理には、ライン・ストレージRAM 318を使用することもできる。通常、ライン・ストレージRAMは、垂直に隣接したパッチのマージに使用される。しかしながら、ライン・ストレージRAMは、パッチの1平面または平面のグループを、オリジナルのパッチ・データそのものとマージするのに使用することもできる。これは、ライン・ストレージRAMが完全なラインを集積し、かつ

いうことである。この回転とシフトとを使用してマスク書き込みを実行すると、様々な効果を得ることができる。

b) 平面スワッピング

本システムおよび方法は、平面を交換およびスワップさせることができる（すなわち、平面間処理）。既述したように、入力パッチ・レジスタ312は、出力レジスタ330とは独立した平面選択を行うことができる。TDM操作では、2平面のスワップが可能である。TDMまたは非TDM走査のいずれかで、いくつかの平面を他の平面のオーバーライトに使用することができる。この操作は、出力レジスタ平面選択をスワップさせるか、または入力平面選択と異なる設定にすることにより実行される。

本システムでは、第1パッチの任意の数の平面を、第2パッチの任意の数の平面でオーバーライトさせることもできる（パッチ間処理）。他のシステムまたはラスタ操作プロセッサ（できればもう

一つのBlitプロセッサ)と本システムおよび方法にシンクロナイズさせることにより、与えられたパッチ平面の代わりに外部ソースからのパッチを、出力マルチプレクサ324を経由して、出力レジスタ330にロードすることができる。通常、出力レジスタは、同じ画像メモリに書き込みを実行する2つのBlitプロセッサに使用することができる。

上記の特徴の応用として考えられるのは、マスク・スワッピング、グラフィック・オーバーレイの変更(すなわち、画像にオーバーレイした1平面で、1サイズのグリッドをスワップし、次に別のサイズのグリッドのスワップを行う)、および画像のエンコーディング/デコーディングである。

IV. 変形と強調

これらの技術には、多くの変形および強調を施すことができる。例えば、一度に完全な8ビットのパッチ平面を処理するために、8台のBlitプロ

セッサ(またはTDM使用の4台)を並行して実行させることができる。この場合、入力レジスタ論理はパッチの各平面が、Blitプロセッサの一つへ送信されるように、また出力レジスタ論理は各Blitプロセッサから平面を受信し1つのパッチを形成するように、変更することができる。同様に、さらに多くの平面を処理する場合は、さらに多くのBlitプロセッサを使用することができる。また、ハードウェアには、一つ以上の適用指定集積回路(Application Specific Integrated Circuit:ASIC)を用いることができる。また、すべての平面を同時に処理することにより、Blitプロセッサは論理演算同様、算術演算を実行することができる。

さらに、もう一つのライン・ストレージRAM(現行ライン・ストレージRAMと同サイズ)を使用し、ページモードで読み込まれたデスティネーション・ラインを保存することにより、論理演算中のデスティネーション読み込みページモード操作の中断を回避することができる。このRAMは、

画面ソース1行が読み込まれた後、直接ロードされる。次に、RAMはライン・ストレージRAMとともに並列に読み込まれる。各RAMは、Blitプロセッサの論理装置316の1入力と接続している。デスティネーション・パッチは、無論デスティネーションに適合しているため、これらのRAMのロード中は、シフトおよびマージのハードウェアは必要でなく、従ってRAMは入力データバス304と直接接続できる。論理ユニット317の第2入力に接続している2つめのライン・ストレージRAMもまた、ソースと第2ソース間の操作を実行でき、デスティネーション(第1および第2ソース以外の)への出力を備えたものであってもよい。この場合、第2ライン・ストレージRAMは、シフトおよびマージの後に接続される。第2ソース・パッチ平面は、デスティネーションと適合させる必要がないからである。

さらなる変形として、シフト/マージのマスク・データの1行を保存する別のライン・ストレージRAMを使用することもできる。これにより、マ

スクコピーの際に、Blitプロセッサは各平面に対し、繰り返しマスク平面処理を行う必要がなくなる。このRAMは、出力パッチ・レジスタへの2つ以上の平面の処理に影響を与えることなく、ブライベート、データバスを通じて、書き込みマスク・レジスタ320に直接読み込むことができる。このRAMを利用する場合、プロセッサは、次の行に進む前に、出力行のすべての平面を処理しておく必要があり、それによってマスク平面は、一度の操作しか必要としないことになる。この場合、ライン・ストレージRAMは、TDMの2面に比べ8面すべてのデータを保存するため、4倍の大きさが必要となる。

また、論理装置316の出力と書き込みマスク・レジスタの入力との間に、第2のALUを使用することができる。この第2のALUの使用により、グラフィックス・プロセッサ308は、マスク・コピーのエッジのマスク・データの"AND"処理を実行する必要がなくなる。このALUの第2入力が、現行エッジ/コーナー・パターンを持つグラフィ

ックス・プロセッサ308のソースとなる。データを書き込みマスク・レジスタに直接バスするためには、ALUには、バイパス・バスまたはモードへのバス設定が必要である。

また、Blitプロセッサは、時間領域多重化方式を用いて8平面すべてを処理するよう修正することができる。これは、Blitクロックの速度をプロセッサ・クロックの8倍にする。(8フェーズをすべてカウントするため)3本のフェーズ表示信号ラインと、処理平面を追加するごとにXシフトおよびマージ論理回路314の追加が必要となる。ライン・ストレージRAM 318もまた、2平面操作の場合の4倍の大きさが必要となる。8フェーズ操作では、バッチの全8平面のシフトおよびマージを最初に行う。次のバッチの平面がバッチ入力レジスタ312から供給されると、Xシフトおよびマージ回路は、第2バッチの各平面に対応する平面とマージする。デコーダPAL 632およびXシフト制御ライン610は、2フェーズTDM操作と同様の処理を実行する。出力バッチ・レジスタ

第3図は、本発明のシステム(アーキテクチャ)の一実施例を示す概略ブロック図、

第4図は、バッチ・フォーマットを表したもので、8平面の5×4バッチを示し、バッチの各平面の画素ポジションに参照番号(0~19)を記した説明図、

第5図は、バッチ(またはバッチ平面)の読み込みおよび書き込みにより、画素をラスタ操作に配列するのに必要な2次元のシフトおよびマージ操作を説明するためのマップを示す説明図、

第6図は、本発明のXシフトおよび/またはマージ機能を実行するための第3図におけるXシフトおよびマージ・ブロック314の一実施例を示した詳細なブロック図、

第7図は、本発明のYシフト機能を実行するための第3図におけるYシフト・ブロック316の一実施例を示した詳細なブロック図、

第8図は、本発明のYマージおよびページモード・アクセス機能を実行するための第3図におけるアドレス・ジェネレータ333およびライン・ス

330はすべてのシフトおよびマージされた平面を集積し、画像メモリに完全なバッチを書き込む。論理装置は、2フェーズTDM操作と同様の処理を実行し、デスティネーションとXYシフト後のソース平面との間のブール演算を実行する。

V. 結論

これまでの説明では、好適実施例および変形例について述べたが、これらは例示であって、本発明の範囲を限定するものでないことを理解されたい。

4. 図面の簡単な説明

第1図は、第1画像のソース領域(SOURCE 1)および第2画像のデスティネーション領域を示したビデオ・ディスプレイ画面を表す説明図、

第2図は、第1図のソース領域から論理演算"OR"を実行して、第1図のデスティネーション領域にブロックコピーを行った後のビデオ・ディスプレイ画面を表す説明図、

トレージRAM 318のブロックの一実施例を示した詳細なブロック図、

第9図は、バッチを保存し、バッチ平面を供給するための第3図における入力バッチ・レジスタ・ブロック312の一実施例を示した詳細なブロック図、

第10図は、第12図示のXシフトおよびマージブロック314、およびその他の種々のシステム・ブロックの実施例に用いられる時領域多重(TDM)モードのクロックおよびデータ信号を示すタイミング・チャート、

第11図は、第12図示のXシフトおよびマージブロック314、およびその他の種々のシステム・ブロックの実施例に用いられる非時間領域多重モードのクロックおよびデータ信号を示すタイミング・チャート、

第12図は、時間領域多重モードに適合する第3図のXシフトおよびマージ・ブロックの他の実施例を示す詳細なブロック図、

第13図は、第3図示の出力レジスタ・ブロック

330 の実施例を示す詳細なブロック図、

第14図は、本発明で実行される書き込みマスク操作を示す説明図、

第15図は、本発明のXシフトおよびマージ機能と方法とに含まれる代表的なステップを示すジェネラル・フロー・チャート、

第16図は、本発明のYシフトおよびマージ機能と方法とに含まれる代表的なステップを示すジェネラル・フロー・チャート、

第17図は、本発明のXYシフトおよびマージ機能と方法とに含まれる代表的なステップを示すジェネラル・フロー・チャート、

第18図は、ソース・パッチとデスティネーション・パッチ間の論理および/または算術演算の代表的なステップを示すジェネラル・フロー・チャートである。

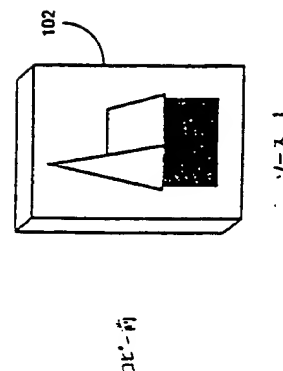
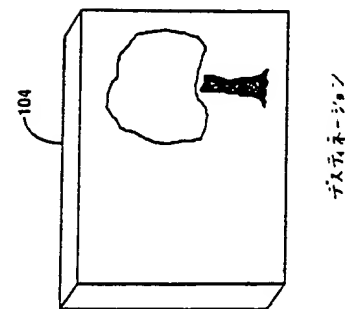
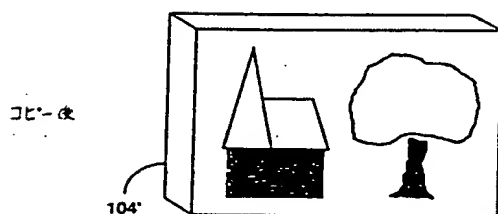


Figure 1

Figure 2



注: デスティネーションは
第1ソースとして使用される

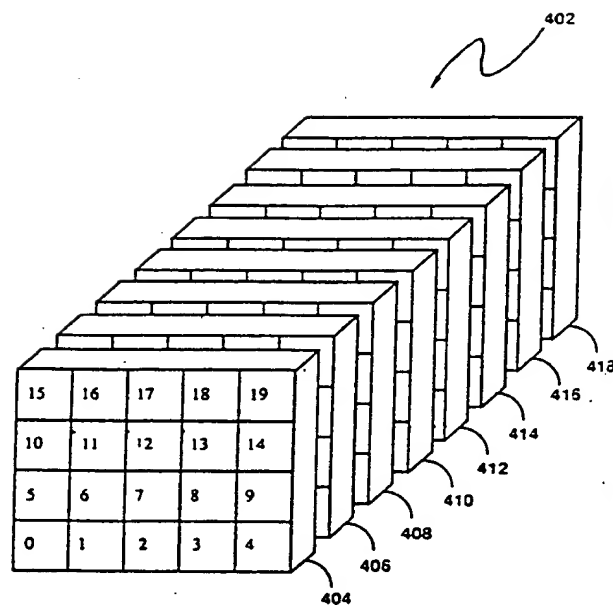


Figure 4

Figure 3

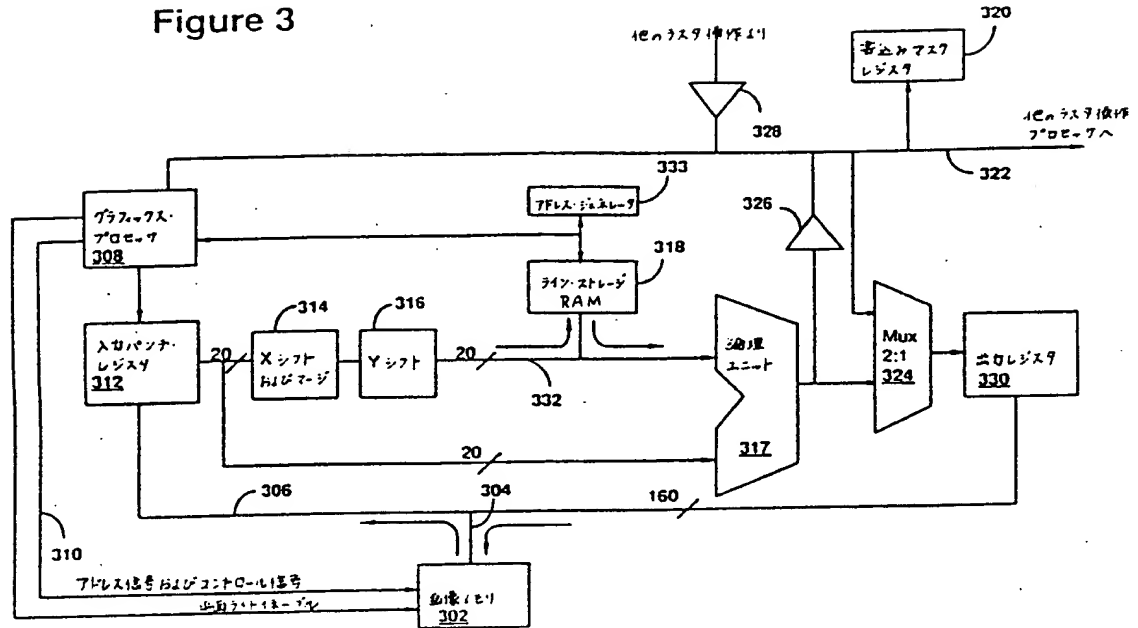
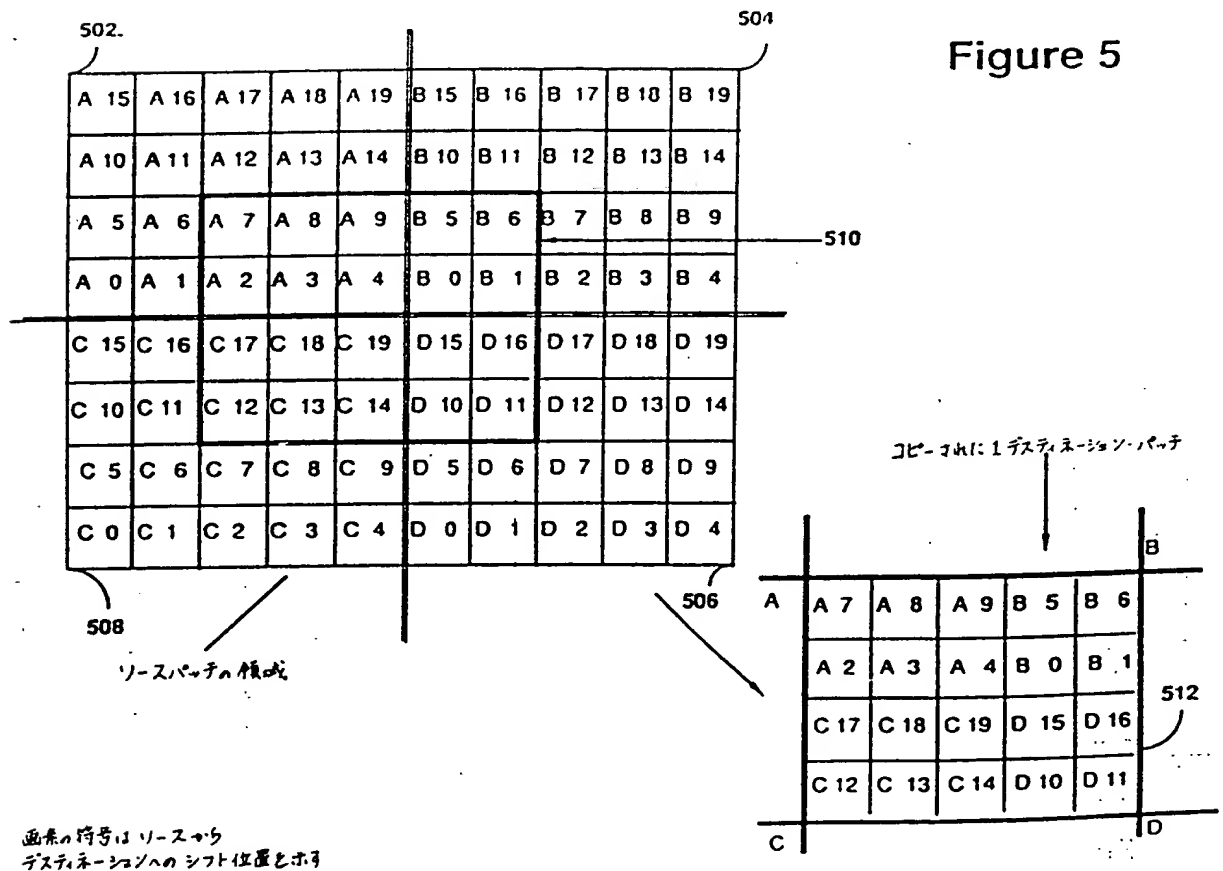


Figure 5



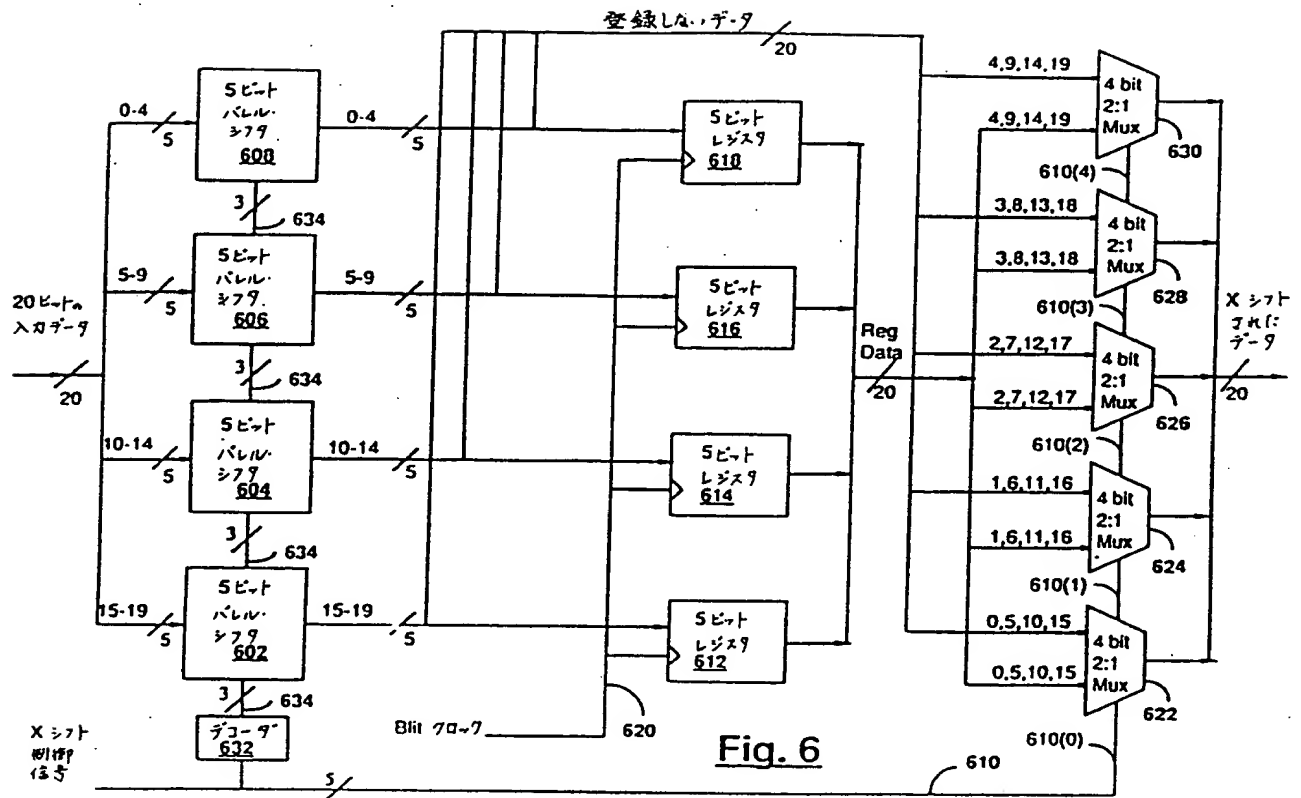


Fig. 6

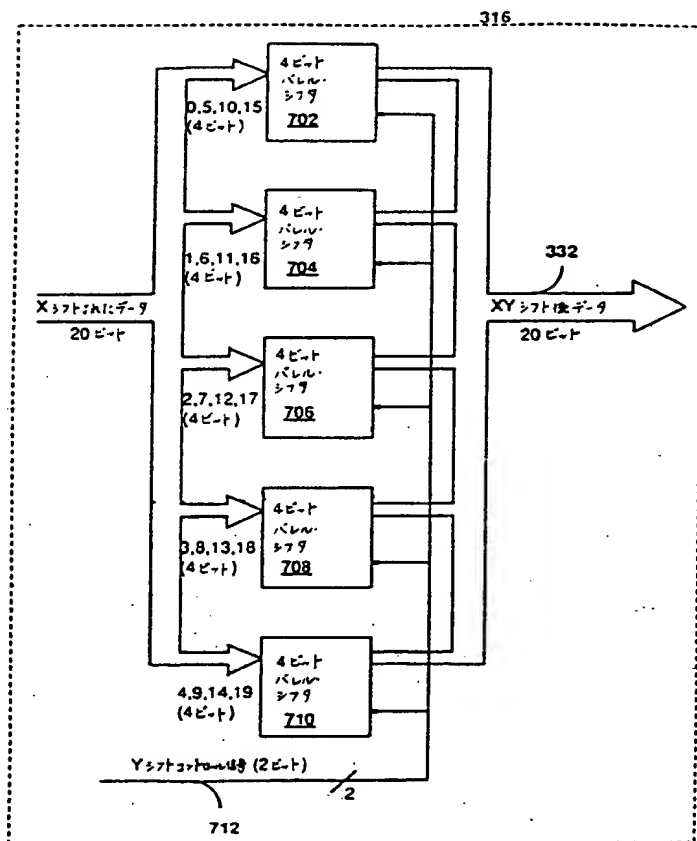


Figure 7

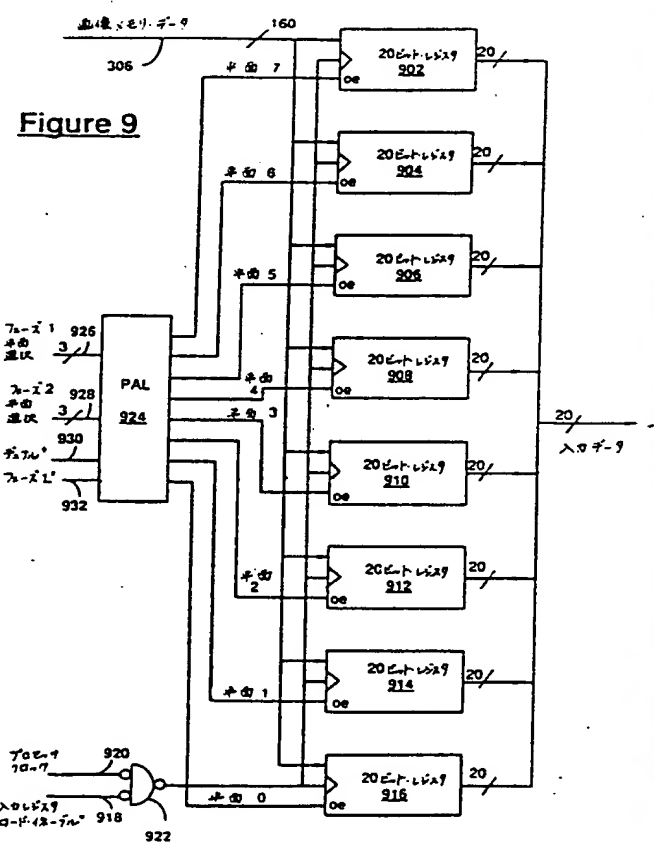


Figure 9

Figure 8

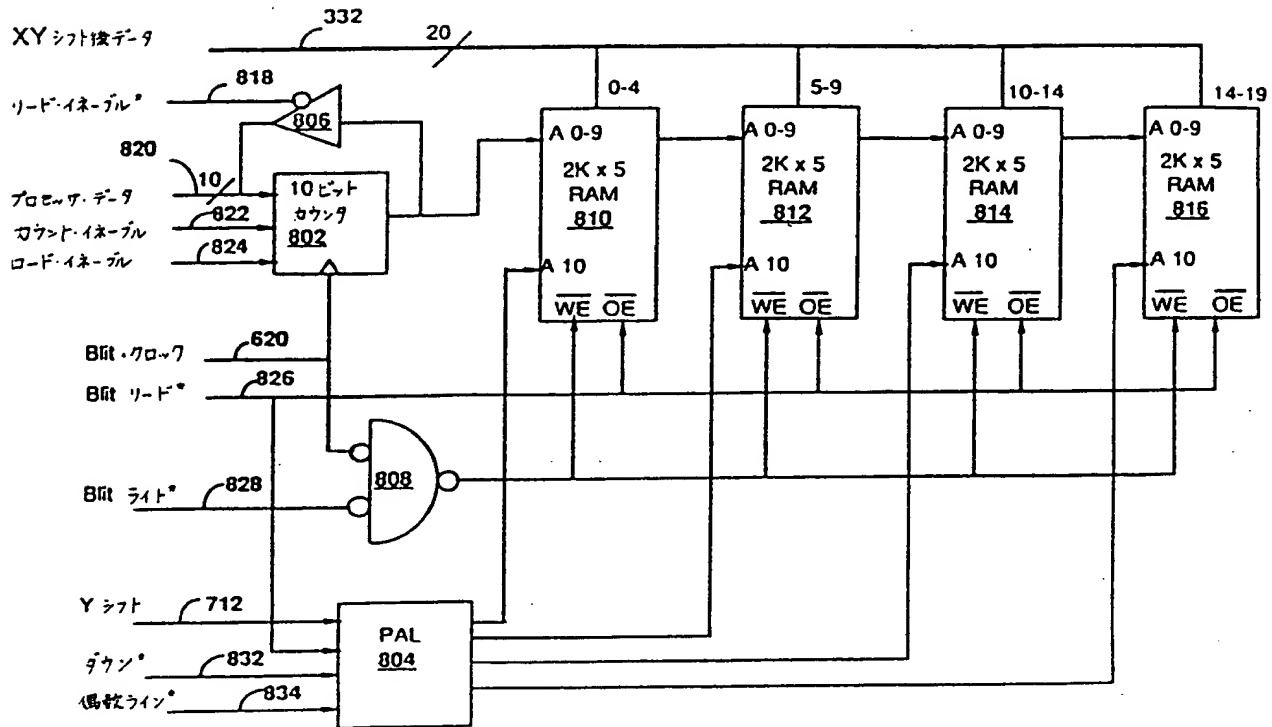


Figure 10

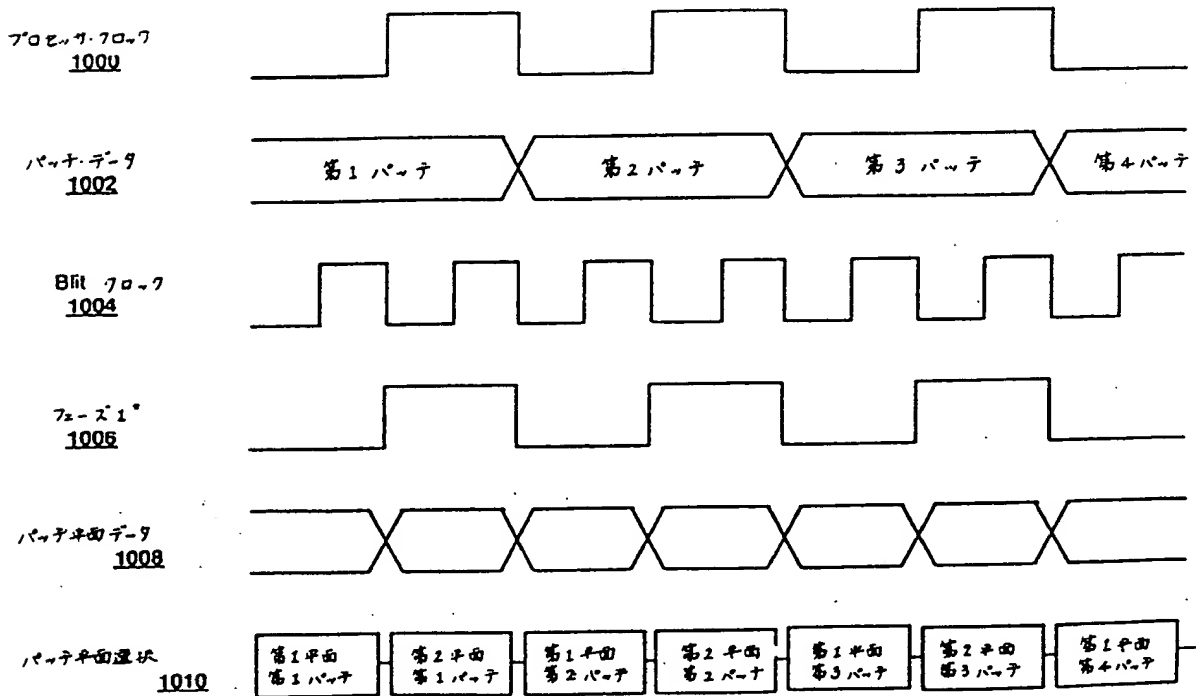


Figure 11

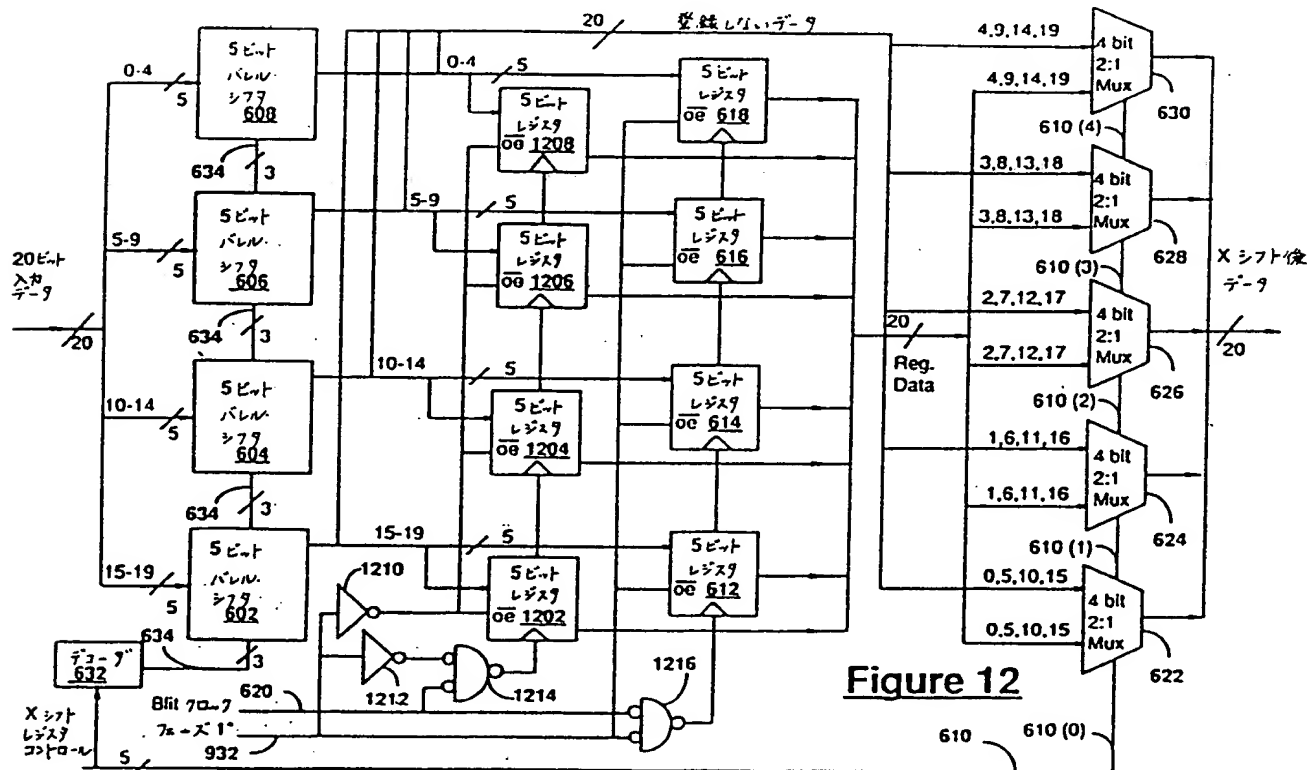
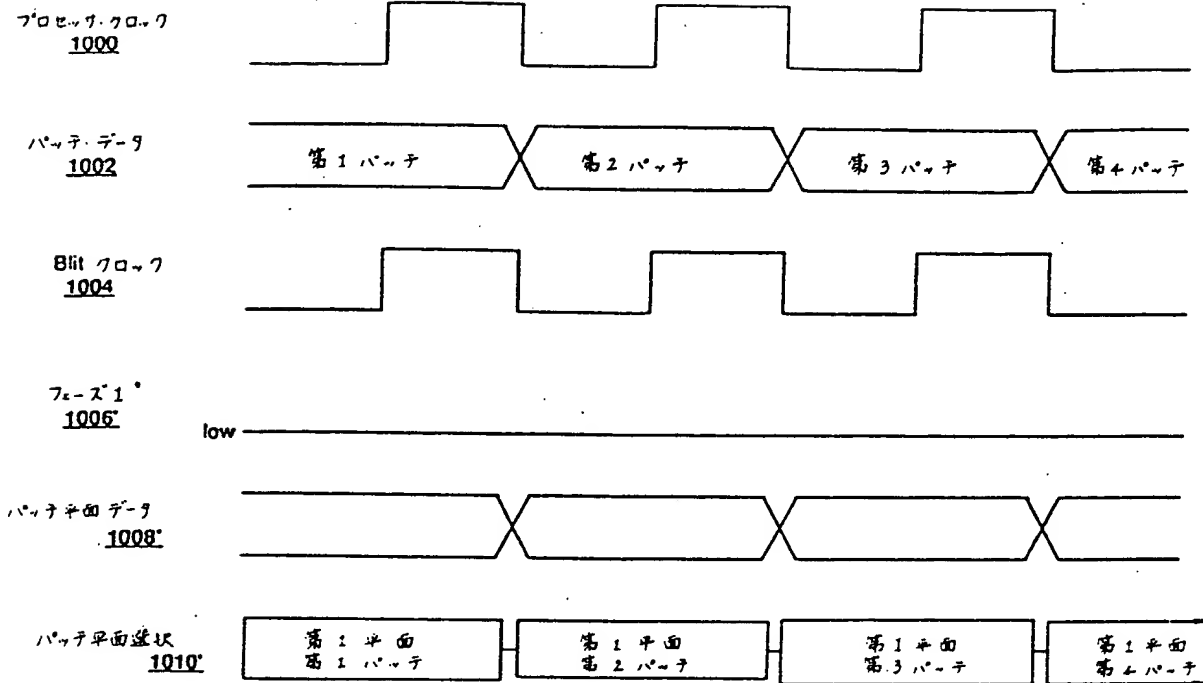


Figure 12

Figure 13

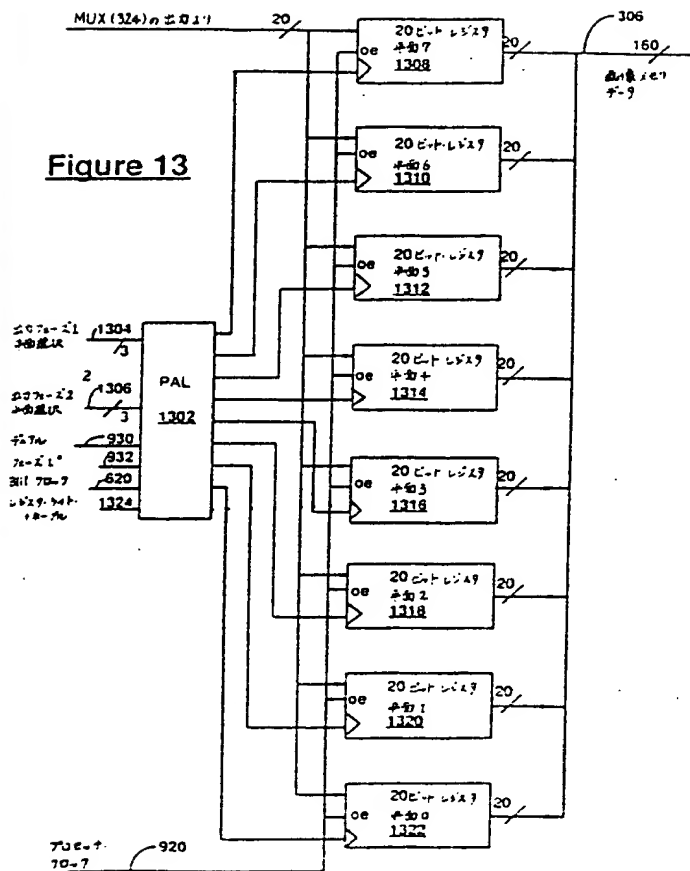


Figure 16

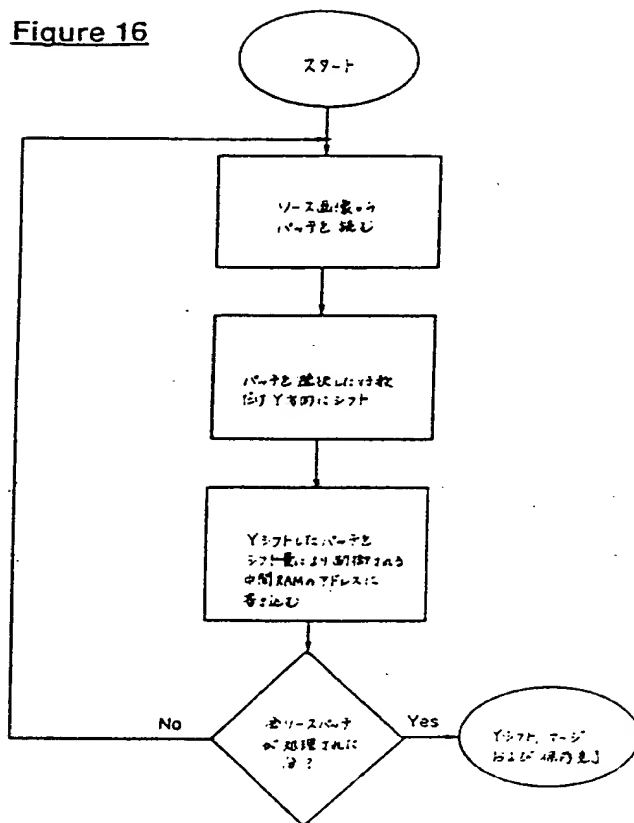


Figure 14

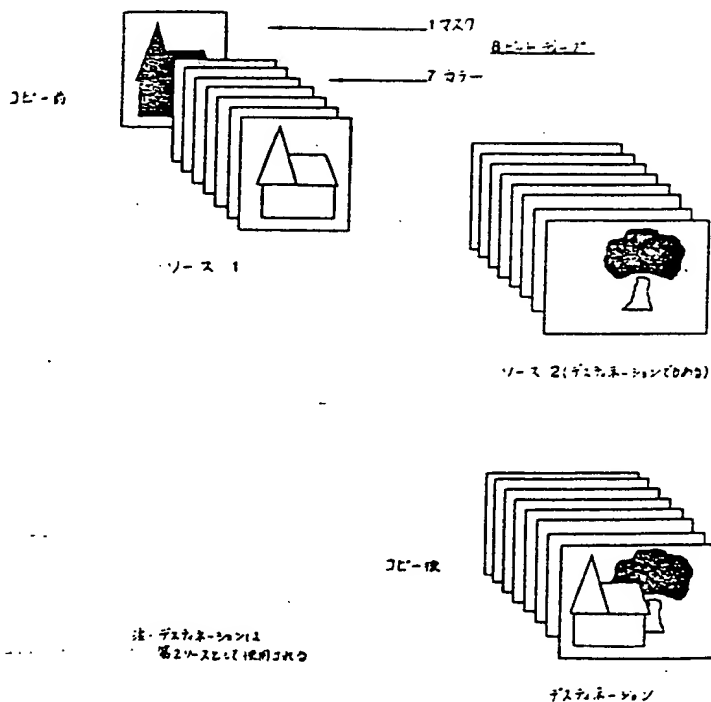


Figure 15

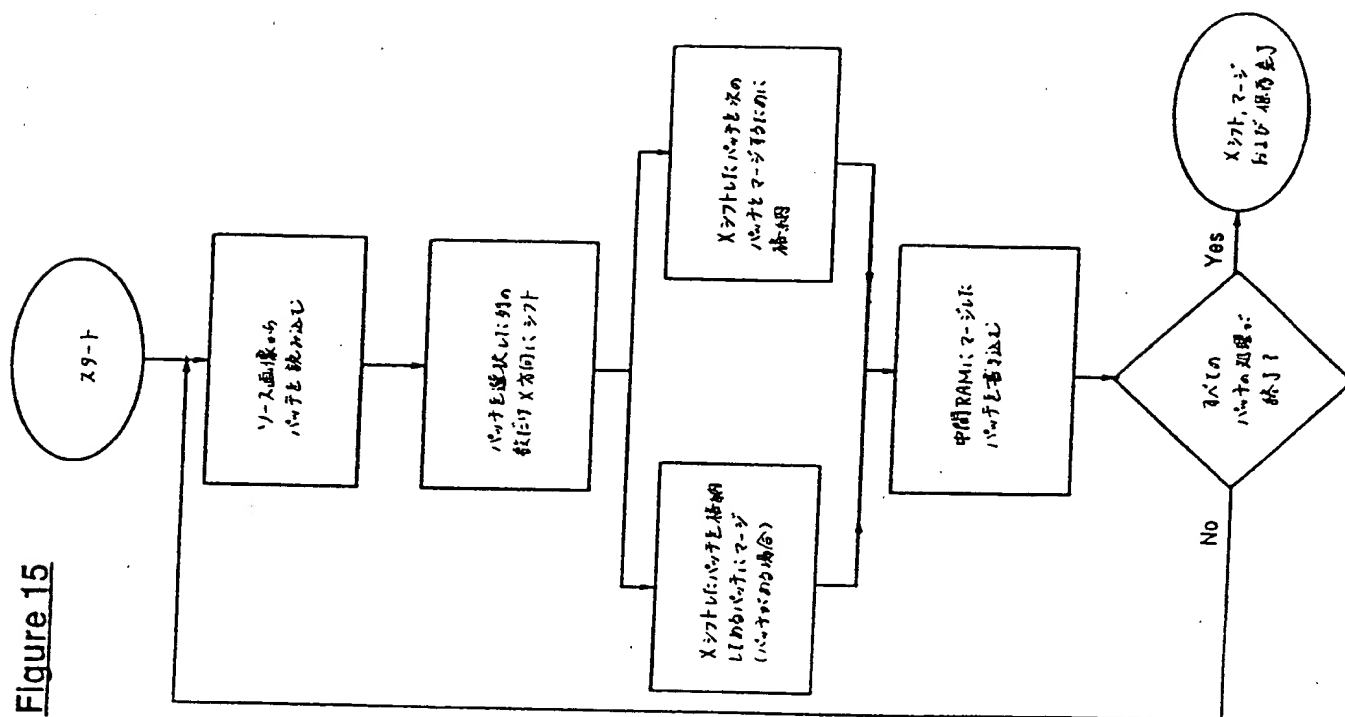
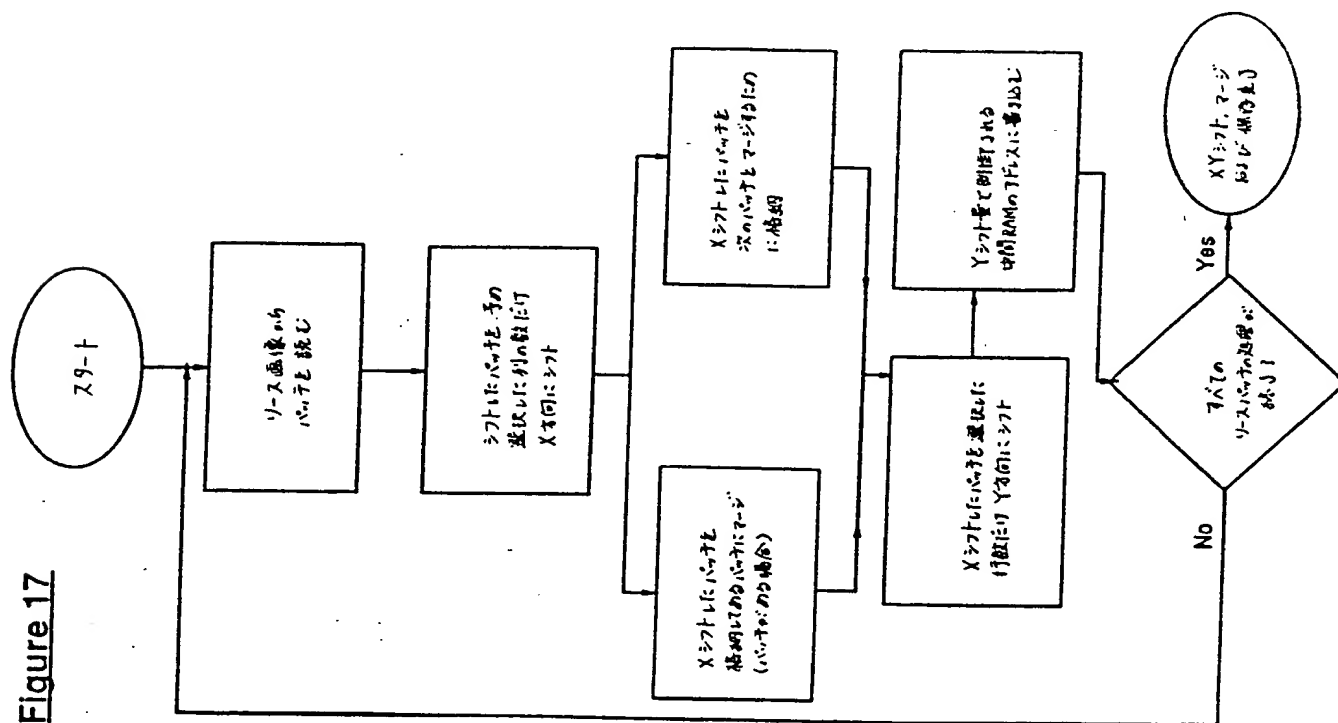
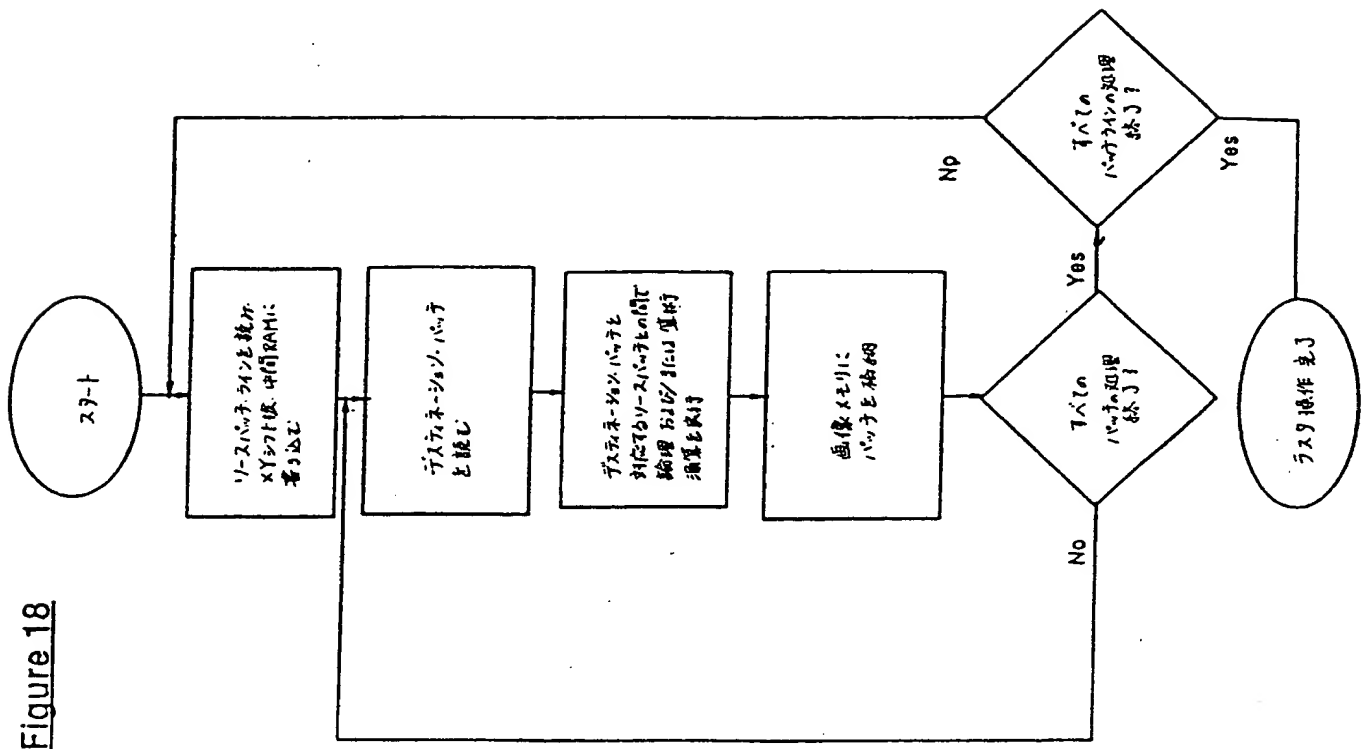


Figure 17





手続補正書 (方式)

平成 1 年 7 月 24 日

特許庁長官 殿

1. 事件の表示

特願平 1-71604号

2. 発明の名称

ラスタ操作実行方法、時間領域多重化方法
および画像処理方法

3. 補正をする者

事件との関係 特許出願人

デュボン ピクセル システムズ リミテッド

4. 代理人

〒107

東京都港区赤坂 5 丁目 1 番 31 号

第 6 セイコービル 3 階

電話 (03) 589-1201 (代表)

(7748) 弁理士 谷 義 一

5. 補正命令の日付

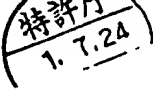
平成 1 年 6 月 12 日 (発送日: 平成 1 年 7 月 4 日)

6. 補正の対象

願書の特許出願人の欄、委任状および図面
の第 6 図および第 12 図

7. 補正の内容

別紙の通り



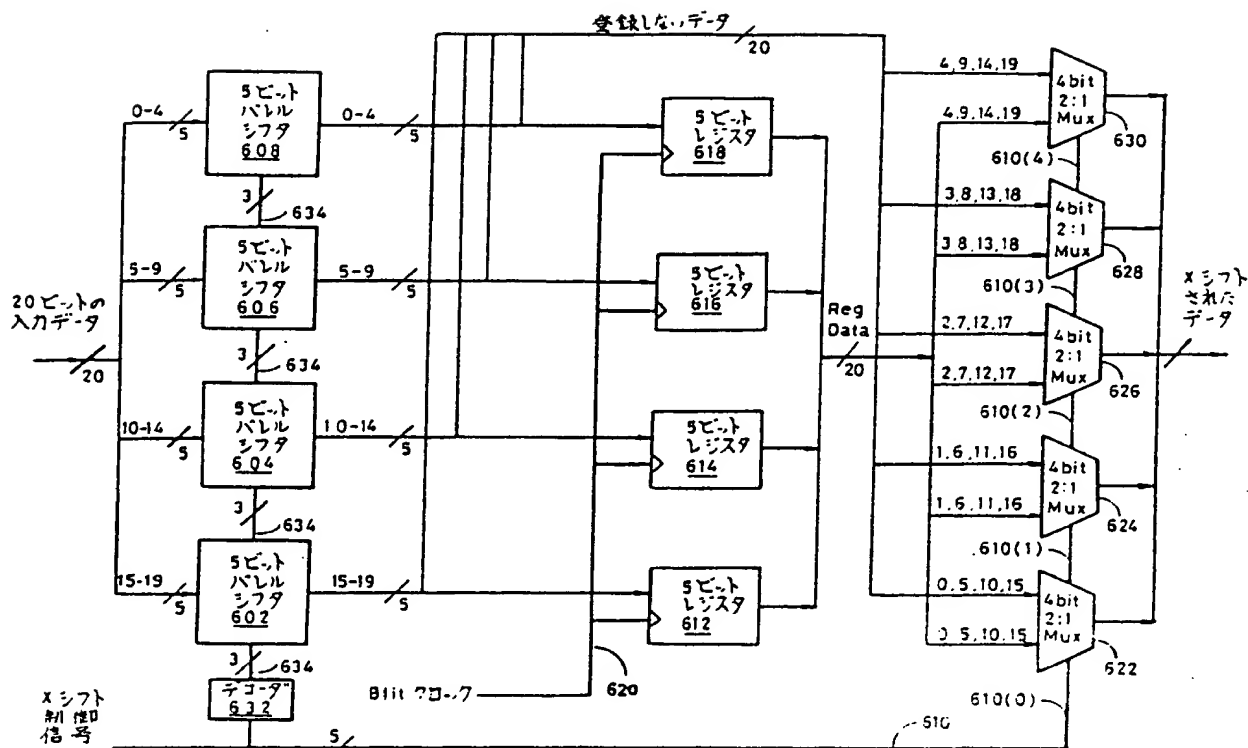


Fig. 6

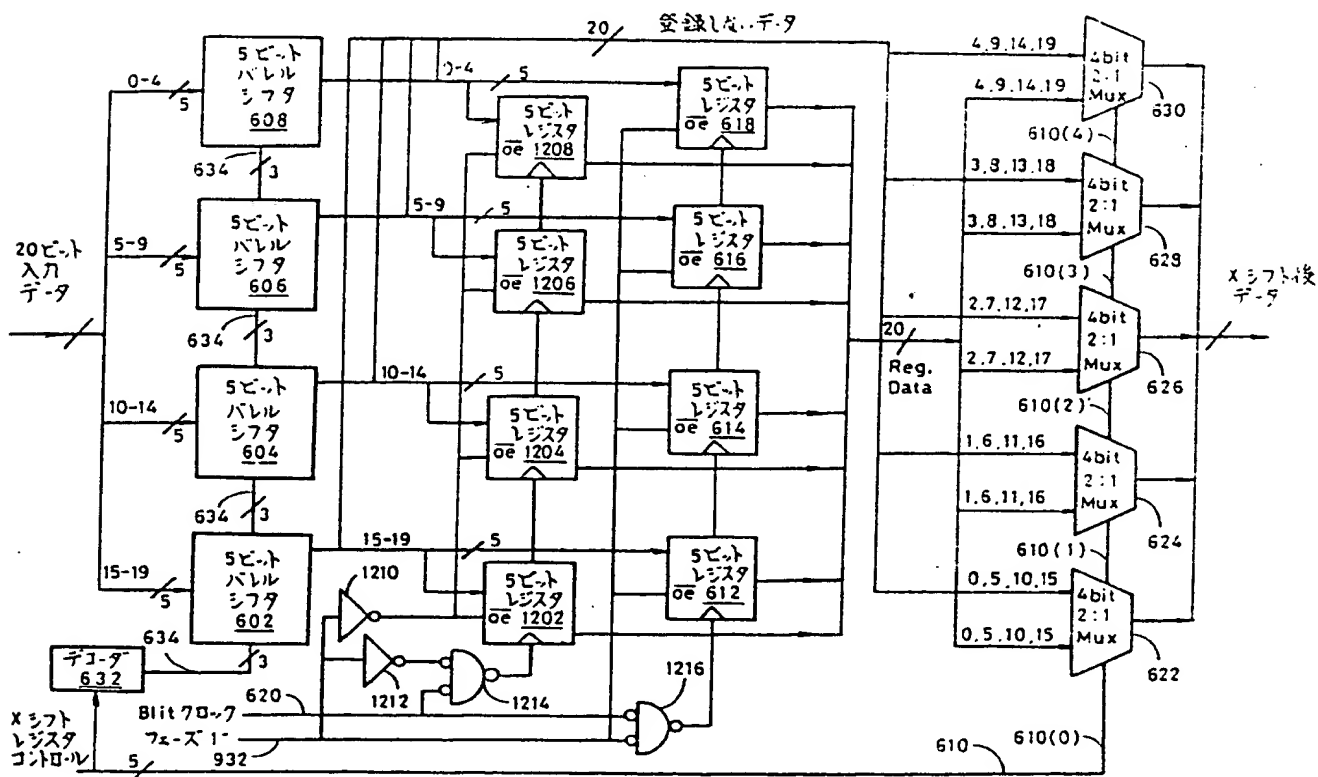
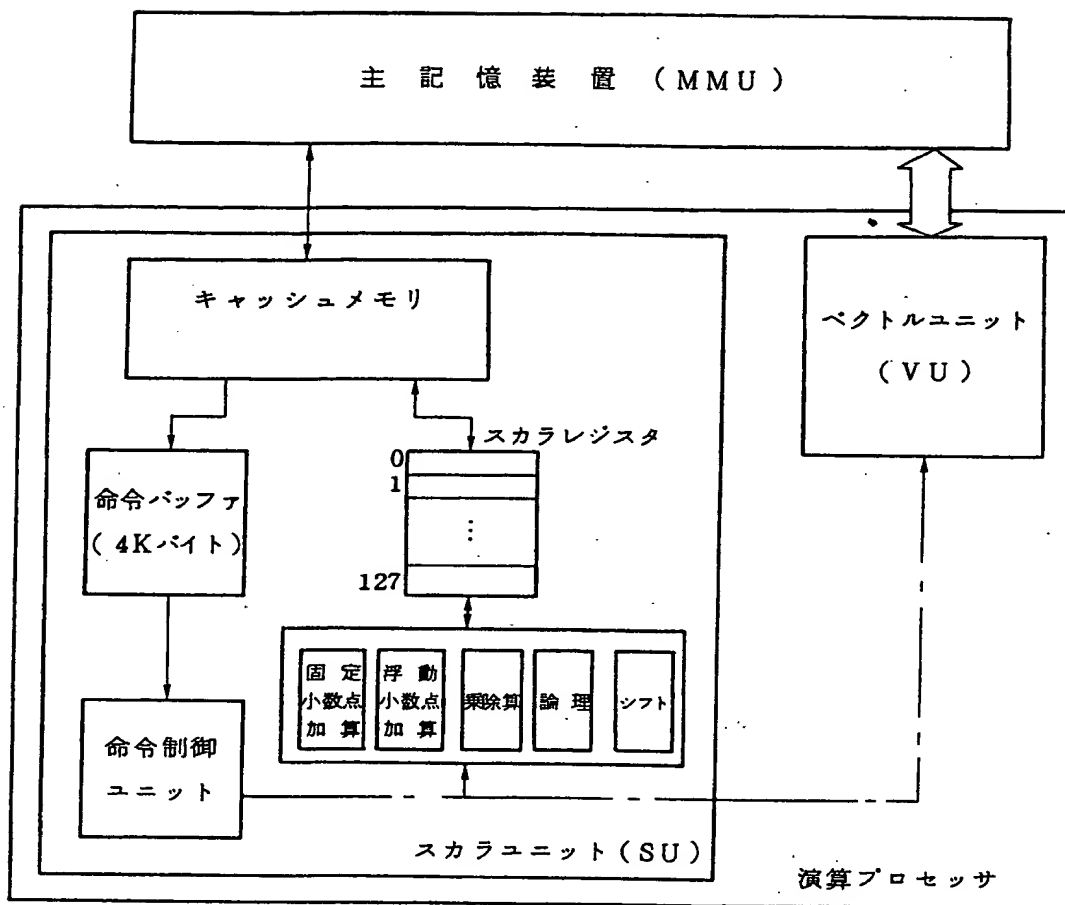


Figure 12

図 7-4 スカラユニットの構成



スカラ処理の性能向上のために、演算プロセッサ (AP) は演算方式として、スカラ演算についても、ベクトル演算同様、演算パイプライン方式を採用しています。

従来、パイプライン制御方式を採用しているコンピュータでも、パイプライン化されているのは、命令の解説、アドレス計算、データ取出しなどの演算実行に必要な前処理部分であって、加算とか乗算などの演算自身は、パイプライン化されていない場合が多くありました。このとき、この演算実行に複数マシンサイクルを要するとすると、図 7-5 (a) に示すように、次の命令の演算実行がその時間だけ遅れることとなります。これに対して、演算プロセッサ (AP) はスカラ演算についても、パイプライン方式を採用しているので、図 7-5 (b) に示すように、レジスタの競合がない限り、演算実行も並列処理されることとなります。